21世纪高等学校规划教材 | 计算机科学与技术



动态网站开发教程（Dreamweaver+MySQL+PHP）

杨秋翔　董晓丽　李雯　贾志燕　编著

清华大学出版社

21 世纪高等学校规划教材·计算机科学与技术

动态网站开发教程
（Dreamweaver＋MySQL＋PHP）

杨秋翔　董晓丽　李　雯　贾志燕　编　著

清华大学出版社
北　京

内 容 简 介

本书以“网站开发基础→MySQL 开发基础→PHP 开发基础→网站开发综合实例”为主线，全面、翔实地介绍了使用 Dreamweaver、MySQL 和 PHP 进行动态网站开发的具体步骤和技巧，涵盖了使用 PHP 进行动态网站开发的主要内容。

本书内容循序渐进、语言简明、容易入门，动态网站初学者可以通过本书的学习，了解并掌握使用 PHP 动态网站开发技术开发实际项目所需要使用到的基本知识。

本书适合于网页制作人员、PHP 网络开发人员和广大网络爱好者学习使用，尤其适合 PHP 动态网站开发初学者，便于了解整个动态网站开发全过程及所用的基本知识和技术。

图书在版编目(CIP)数据

动态网站开发教程：Dreamweaver＋MySQL＋PHP/杨秋翔等编著.--北京：清华大学出版社，2015(2016.8 重印)

21 世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-41996-9

Ⅰ. ①动…　Ⅱ. ①杨…　Ⅲ. ①网站－设计－高等学校－教材　Ⅳ. ①TP393.092.1

中国版本图书馆 CIP 数据核字(2015)第 263188 号

责任编辑：郑寅堃　赵晓宁
封面设计：傅瑞学
责任校对：焦丽丽
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：虎彩印艺股份有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：22　　**字　　数**：536 千字
版　　次：2015 年 11 月第 1 版　　**印　　次**：2016 年 8 月第2 次印刷
印　　数：1001～1300
定　　价：45.00 元

产品编号：065357-01

出 版 说 明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”)，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程(群)加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”)，旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性(新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐)，经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括：

(1) 21世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

(2) 21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21世纪高等学校规划教材·信息管理与信息系统。

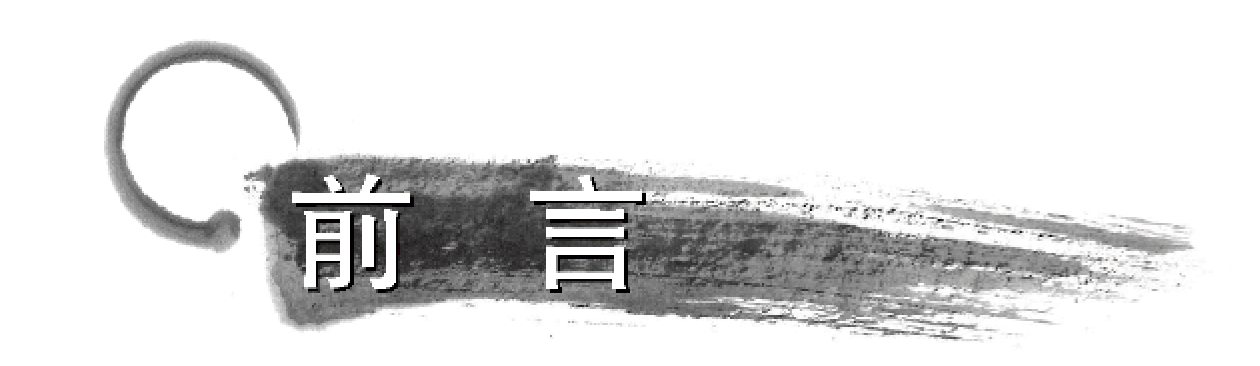
(6) 21世纪高等学校规划教材·财经管理与应用。

(7) 21世纪高等学校规划教材·电子商务。

(8) 21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
联系人：魏江江
E-mail:weijj@tup.tsinghua.edu.cn



前言

PHP 是一个操作简单、功能强大的动态网站开发语言，用 PHP 开发的网站，开发过程简单，用户维护方便。由于具有很好的安全性、跨平台性及开源性，PHP 自出现以来，得到了广泛的应用和迅速的发展。在国内，PHP 的发展也非常迅速，百度、新浪、搜狐等各大互联网门户都在广泛使用 PHP，越来越多的小门户网站也加入使用 PHP 的行列。根据调查，PHP 从业者选择 PHP 的因素主要是“语言简单容易上手”、“可获得大量开源代码”和“PHP 免费”三个方面，今后的发展潜力巨大，更多的年轻人对 PHP 产生了浓厚的兴趣。

学习 PHP 动态网站开发技术，前提必备基础知识包括静态网站制作基础(HTML 基础知识、Dreamweaver 使用方法等)、数据库技术。通过多年的教学实践，分析学生在学习过程中普遍存在的问题，作者认为对 PHP 动态网站开发有浓厚兴趣的学习者，选择通俗易懂且容易入门的教材，有助于学生了解并掌握整个动态网站开发全过程及所用的基本知识和技术。鉴于此，通过整合 HTML 基础知识、Dreamweaver 的使用、MySQL 数据库及 PHP 动态网站知识等内容，使得动态网站初学者可以快速地了解并掌握使用 PHP 动态网站开发技术开发实际项目的基本技能。

全书共 12 章，第 1 章网页制作基础知识，主要介绍网页与网站的基本概念，计算机网络的相关概念以及网页制作相关技术。第 2 章介绍 HTML 基础知识。第 3～5 章为 Dreamweaver CS6 的使用：第 3 章介绍 Dreamweaver CS6 基本操作，包括如何创建站点、制作简单网页、创建链接、表单的使用；第 4 章介绍了 CSS 样式和网页布局规划，主要介绍在 Dreamweaver 中如何使用 CSS 样式，利用表格、框架以及 CSS 对网站页面进行布局；第 5 章为 Dreamweaver 的高级应用，介绍了如何在页面中添加行为、菜单栏、折叠式面板以及 JavaScript 简单应用等。第 6～8 章为数据库基础：第 6 章介绍数据库的基本概念、数据模型、关系模型等数据库基础知识；第 7 章主要介绍 MySQL 服务器的安装、配置及使用；第 8 章详细介绍 MySQL 数据库的基本操作，包括使用 SQL 语句与客户端界面工具如何定义数据库、表、修改表、表中数据增删改查操作以及 MySQL 数据库的备份与恢复。第 9～12 章为 PHP 技术内容：第 9 章介绍 PHP 基本知识，包括基本语法、数据类型、运算符与表达式、流程控制以及函数等；第 10 章介绍了 PHP 处理表单数据；第 11 章介绍了 PHP 操作数据库，包括连接数据库、创建数据库、建立表、表中增删改选取数据等；第 12 章为 PHP 动态网站开发实例，主要以网上商城管理系统为例，介绍了数据库表结构的设计以及 PHP 实现核心代码。本书围绕 PHP 动态网站开发所需要掌握的基础知识循序渐进展开，通过这种安排，力图使学生掌握使用 PHP 技术进行动态网站开发的基本技术方法。

本书语言简明、内容丰富、容易入门，动态网站初学者可以通过本书的学习，了解并掌握使用 PHP 动态网站开发技术开发实际项目所需要使用的基本知识，有静态网页基础知识的学习者可选择 MySQL 数据库、PHP 动态网站开发、综合案例等部分内容进行学习。选用该教材的教师可以参考已修课程，选择其中的内容进行教学。

本书第 2 章由杨秋翔编写,第 5～8 章由董晓丽编写,第 1 章和第 9～12 章由李雯编写,第 3、4 章由贾志燕编写,全书由董晓丽最后统稿。

本书的编写参考了大量的相关书籍和资料(详见参考文献),在此向这些参考文献和书籍的作者表示感谢。

本书在编写过程中得到了中北大学软件学院、计算机与控制工程学院领导以及同事的大力帮助和热情支持,在此向他们一并表示感谢。

由于编者水平有限,书中难免存在疏漏之处,欢迎广大读者和同行专家,尤其是使用本书的教师和学生批评指正。

编　者

2015 年 6 月

目录

第1章 网页制作基础知识

本章要点：

- 网页与网站基本概念；
- 计算机网络相关概念；
- 网页相关技术；
- 网站分类与网站托管。

随着通信技术及互联网技术的迅猛发展，网络应用已成为人们工作、学习、生活不可缺少的一部分，越来越多的政府部门、企事业单位、学校以及个人通过建立网站发布信息，信息的传递主要依靠计算机网络，在学习如何制作网页之前，首先需要了解一些网页与网站基本概念、计算机网络的相关概念。

1.1 网页与网站基本概念

计算机网络是现代社会传播信息的重要途径，而网页与网站承载着各种各样的信息，是人类传递消息在计算机网络中最广泛的应用。可以说，在网上冲浪已经成为人类一种最普遍娱乐活动的今天，网页对人们来说不再是一个陌生的概念，网页已经成为人们与外界沟通的重要桥梁，在信息化生活中，人们无时无刻不在浏览网页，关注对自己有用的信息。

1.1.1 网页基本概念

网页(Web Page)，从字面上理解是在网络上传输的一个页面，这个页面包含大量信息。在多媒体信息化时代，网页用来传送消息的方式不再局限于文本和图片，还可以用动画、声音、视频等多种方式图文并茂地描述信息，使人们身临其境地感受信息。

网页是存储在 Web 服务器上，通过 Web 进行传输，并被浏览器所解析和显示的文档类型。网页是 Internet 上的信息携带者，是 Internet 中最基本的信息单位，是把各种多媒体信息相互链接起来而构成的一种信息表达方式。与人们生活密切相关的如图 1.1 所示的购物网页，与在校学生相关的如图 1.2 所示的学校网页。



图 1.1 购物网页示例



图 1.2 学校网页示例

1.1.2 网站基本概念

通过超链接连接起来的一系列逻辑上可以视为一个整体的若干网页就叫做网站(Web Site),也就是同一类信息内容的集合。简单地说,网站是若干网页的集合,人们可以通过网站来发布自己想要公开的资讯,或者利用网站来提供相关的网络服务,如图 1.3 所示的学校

网站。人们可以在购物网站上买东西，可以在求职网站上找工作，可以在新闻网站上看咨询，可以在个人网站中表达自己的观点。



图 1.3　学校网站示例

网站是表达相关信息内容网页的集合。所以网站中的信息内容是由一个个相互链接的网页，通过网络传送到读者眼前。而网页的传送会涉及一些计算机网络规则与概念，例如因特网、HTTP、IP 地址、域名、HTML 等相关概念，也会用到一些相关的工具软件，例如 Dreamweaver 软件。所以学习网页与网站制作之前，需要理解与网页有关的一些计算机网络相关概念，对制作网页会有一定的帮助。图 1.4 所示是一个动态网站的示例。

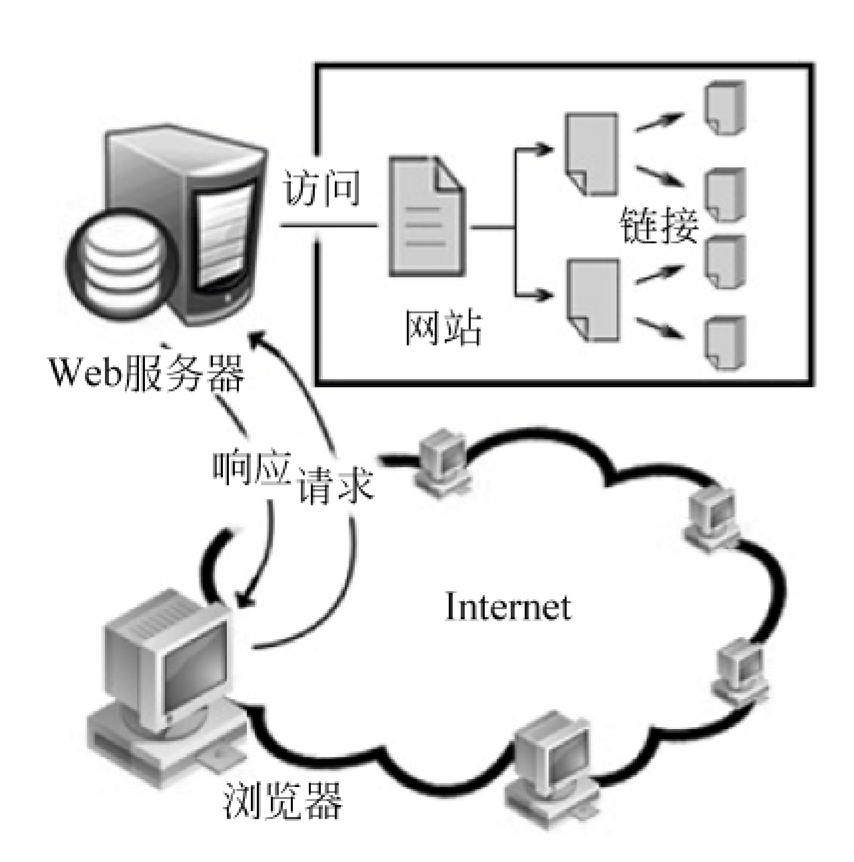


图 1.4　动态网站示例

1.2 计算机网络相关概念

1.2.1 Internet 基本概念

计算机网络是多台计算机通过特定的连接方式构成的一个计算机的集合体，是Internet的基础。

Internet是最著名的广域网，是全世界的计算机相互连接而形成一个整个地球规模的计算机网络，该网络能够将全世界的计算机连接在一起，是因为全世界都遵循一个通信协议——TCP/IP协议，这样，在Internet中的任意一台计算机都可以从Internet上的其他计算机中来获得信息。

Internet起源于1969年，是由美国国防部设立的名为ARPAnet的军用网络。1982年，美国军方从ARPAnet撤出，主要用于学术机构的连接，科学家可以用它来分享数据。今天的Internet，已成为千万计用户使用的全球性网络，没有人或机构拥有或全面控制它。

从国内Internet的发展现状来看，1994年4月，中国科学院计算机网络信息中心正式接入Internet，从此，中国Internet发生了飞速的发展。从域名、网站数、IP地址等增长情况来看，我国互联网资源得到了全面提升。

1.2.2 Internet 基本服务功能

Internet的基本服务功能有很多种，常见的是WWW服务、电子邮件(E-mail)服务、文件传输服务(File Transfer Protocol，FTP)、远程登录服务(Telnet)等。

1. WWW 服务

WWW(World Wide Web，万维网)是一个基于超级文本的信息查询工具。WWW是在Internet的基础上，由各计算机节点上的WWW软件和超级文本格式的信息文件组成的。这些节点称为WWW服务器，简称Web服务器，也就是网站。海量的信息被存储于每一个Web服务器上，也就是网页的集合，用户使用WWW时只需提出查询要求，到何处查询及如何查询则由WWW自动完成。

WWW服务是Internet提供的最主要的服务，通常上网通过浏览器访问到的网页就是万维网的内容。WWW服务工作方式如图1.5所示。

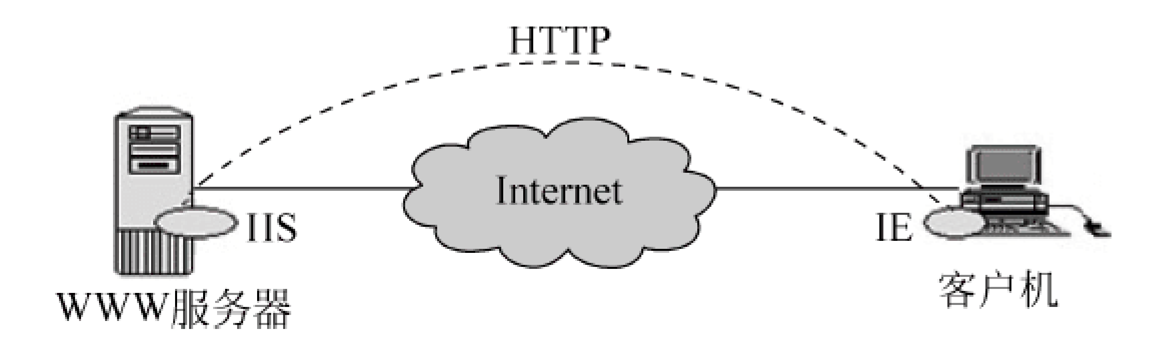


图1.5 WWW服务工作方式

WWW服务由三部分组成，即：客户机、服务器和HTTP协议。

WWW工作过程如下。

(1) 客户机运行 WWW 程序,一般是浏览器,连接在 Internet 上的某个服务器,提出请求。

(2) 服务器响应,发送超媒体文件,一般是由 HTML 编写的网页内容。

(3) 客户机收到信息,通过浏览器解释执行 HTML 将网页内容显示在客户面前。

2. 其他主要服务

电子邮件(E-mail)服务：Electronic Mail 是计算机用户交换信息的一种途径。邮件地址由用户名和主机名组成,中间由@连接,如 zbdx@nuc. edu. cn,主要遵循 STMP 和 POP3 协议,电子邮件服务主流软件是 Foxmail 软件与 Outlook 软件。

文件传输服务(File Transfer Protocol,FTP)：文件传输服务用于在 Internet 上把文件从一个主机传送到另一个主机上。主要遵循 FTP 协议,文件传输服务主流软件有迅雷等。

远程登录服务(Telnet)：远程登录用于用户通过计算机网络进入和使用远程计算机系统。主要遵循 Telnet 协议,利用 DOS 窗口操作。

1.2.3　TCP/IP 协议簇

网络协议是关于信息格式及信息交换规则的描述,或者说是网络中的设备相互“打交道”时共同遵循的一套规则。只有大家都遵守制定的这些规则,在 Internet 上的信息才能畅通无阻地传输。

TCP/IP 是 Internet 上通用的网络标准协议簇,是上百个各种功能协议的总称。前面所提到的 HTTP 协议、FTP 协议等都属于 TCP/IP 协议簇。TCP(Transmission Control Protocol,传输控制协议)提供面向连接的服务,IP(Internet Protocol)是互联网协议。在 Internet 中,应按照相应的协议编写相应的信息进行传输数据,只有按照协议规定,才能保证数据完整传输。

TCP/IP 协议传输过程,采用分组交换的通信方式,在发送端,将信息数据按照一定的格式规则大小封包,每个数据包编上序号,在接收端,利用软件把数据还原成用户能够读懂的信息文件。

在数据包传输的过程中,必须知道发送主机与接收主机的地址,才能确保数据包发送成功。IP 协议负责记录发送主机和接收主机的地址,在 Internet 上每一台主机及网络设备都必须有唯一标识。IP 地址就像 Internet 上每一个主机和网络设备的门牌号。IP 地址是一个 32 位的二进制数组成的号码,该号码分为 4 段,每段 8 位,每段取值范围在 0～255 之间。

IP 地址有两种表示方法：二进制和十进制。

例如,中北大学的 IP 地址如下。

十进制：202. 207. xxx. xxx

二进制：11001010. 11001111. xxxxxxxx. xxxxxxxx

1.2.4　域名

IP 地址是由数字表示的比较难记,所以必须用一种新方法来代替这种数字,即“域名”地址。域名由几个英文单词组成；Internet 域名是 Internet 网络上一个服务器或一个网络

系统的名字,在全世界没有重复的域名。

如果要访问中北大学的网站,用户可以在浏览器的地址栏中,输入 http://202.207.xxx.xxx 打开中北大学主页。访问中北大学时,也可以不用使用 IP 地址,在浏览器的地址栏中输入 www.nuc.edu.cn 就可以访问该网站。而 www.nuc.edu.cn 具有一定的意义,其中 cn 代表中国,edu 代表教育网,nuc 代表中北大学,www 代表全球网,整个域名合起来就代表中国教育网上的中北大学站点。

域名易于记忆,但是网络通信中要使用二进制,所以在传送信息之间只能标识 IP 地址。对于 IP 地址与域名的关系可以用一台专门的服务器——DNS 服务器记录与解析。

域名的表示方法:

注册域名.二级域名.顶级域名

(1) 顶级域名:分为组织机构(ac 科研机构,com 工、商、金融等企业,edu 教育机构,gov 政府部门,net 互联网络、接入网络的信息中心和运行中心,org 各社会团体及民间非营利组织)和地理模式(cn 中国,jp 日本)。

(2) 二级域名:由该国自行确定,分为类别域名和行政区域名(sx 山西,sh 上海)。

(3) 注册域名:在互联网中申请注册的名字,例如 nuc.。

1.2.5 DNS 服务器

DNS(Domain Name System 或 Domain Name Service,计算机域名系统)是由域名解析器和域名服务器组成的。一般来说,在大型的局域网中,都会配备一台 DNS 服务器。

DNS 服务器的任务就是自动将域名转换为对应的 IP 地址。因为虽然域名易于记忆,但是在 Internet 中网络通信只能标识 IP 地址。

域名系统采用类似目录树的等级结构。域名服务器为客户/服务器模式中的服务器方,它主要有两种形式:主服务器和转发服务器。将域名映射为 IP 地址的过程就称为“域名解析”。

1.2.6 浏览器

浏览器是指将互联网上的文本文档(或其他类型的文件)翻译成网页,并让用户与这些文件交互的一种软件工具,主要用于查看网页的内容。浏览器有许多种,按其运行的平台分类,主要有基于 UNIX 系统、Microsoft Windows 系统和 Apple Macintosh 系统三种类型。目前最常用的浏览器有:美国微软公司的 Internet Explorer 和 Google 公司的 Chrome、Mozila 提供的 Firefox 等。早期曾经很出名的美国网景公司的 Netscape Navigator 现在已经很少人用了。如图 1.6 所示是市面上常见的浏览器图标。



图 1.6 浏览器图标示例

1.2.7　URL

URL(Uniform Resource Locator,统一资源定位器)也就是网络地址,是在 Internet 上用来描述信息资源,并将 Internet 提供的服务统一编址的系统。简单来说,通常在浏览器地址栏中输入的网址就是 URL 的一种。

URL 主要用来指明通信协议和地址的方式,以取得网站提供的各种服务,URL 格式如下:

```
<通信协议>: //<主机名>/<文件路径>/<文件名>
```

即:

```
Protocol://host.domain/path/filename
```

其中,通信协议包括 HTTP、FTP 等协议。主机名指服务器在网络中的 IP 地址(如 210.77.35.178)或域名(如 www.nuc.edu.cn)。在路径中,主机名与路径文件名之间以"/"分隔。例如:http://www.nuc.edu.cn/index.php/。

1.2.8　超链接

超链接(Hyperlink)是从一个页面跳转到另一个页面的入口,它可以看作是一种文件的指针,提供了相关联文件的路径,并且可以跳转执行相应的文件。

超链接技术是从一个网页指向另一个地址的链接,如指向另一网页或同一网页上的不同位置。链接地址通常是另一个网页,也可以是一幅图片、一个邮件地址等。

网站内部链接,是指本网站网页间的链接。内部链接一般用相对路径。相对是指按当前所编辑的网页所在目录位置来判断其他文件所在的位置,必须在同一盘符内。

网站外部链接,是指链接到其他网站的链接。

1.3　网页相关技术

1.3.1　网页的本质——HTML

HTML(HyperText Markup Language,超文本标记语言)是表示网页的一种标准。HTML 主要包含两个元素:标识符和属性。通过在文本文档中添加标记符和属性,控制简单的文本文档的格式,使用浏览器就能显示出网页。

HTML 元素(Element)构成了 HTML 文件。HTML 文件是一种包含很多标签(Tags)的纯文本文件。

HTML 文件可以使用 Windows 系统的"记事本"或者其他的文本编辑器就可以编辑,HTML 文件以.html 或.htm 为扩展名才会被浏览器"认识"并"解读"出来。

【例 1.1】 网页示例。

```
<html>
<head>
<title>你好!</title>
</head>
<body bgcolor="yellow">
<p>哈哈,我的<strong>第一个</strong>页面!</p>
</body>
</html>
```

1.3.2 网页修饰技术——CSS

随着技术的进步,在网页制作过程中使用 CSS 技术已经成为基本要求。实际上,较新版本的 Dreamweaver 已经将 CSS 技术作为默认的网页修饰技术。

CSS(Cascading Style Sheet,层叠样式表)技术是一种格式化网页的标准方式,它扩展了 HTML 的功能,使网页设计者能够以更有效的方式设置网页格式。

1.3.3 客户端脚本语言——JavaScript

JavaScript 是由 Netscape 开发的一种解释语言,它可以直接插入到 HTML 文档中,用来给 HTML 网页增加动态功能。

JavaScript 是一种基于对象和事件驱动并且有安全性能的脚本语言。使用它的目的是与 HTML 一起实现在一个 Web 页面与 Web 客户交互。比如,利用 JavaScript 语言编程实现客户端的注册页面的数据校验等。

1.3.4 服务器端脚本语言——PHP,ASP,JSP

PHP 是一种开源脚本语言,其语法吸收了 C 语言、Java 和 Perl 的特点,利于学习,使用广泛,主要用于 Web 开发领域,适用于动态网页开发,处理表单,与数据库交互,数据校验等内容。

ASP、JSP 与 PHP 在开发动态网页的功能上类似。ASP 是微软公司开发的服务器端脚本语言,JSP 是由 Sun 公司倡导,许多公司参与一起建立的一种动态网页技术标准。

1.3.5 数据库——MySQL,Access

数据库是依照某种数据模型组织起来并存放在二级存储器中的数据集合。MySQL 是一种关联数据库管理系统,体积小,速度快,总体拥有成本低,开放源码,搭配 PHP 语言和 Apache 服务器软件可组成良好的动态网页开发环境。

Microsoft Office Access 是由微软发布的关系数据库管理系统,一般与 ASP 语言与 IIS 服务器软件搭配开发。

制作网页的简单技术分类如图 1.7 所示。

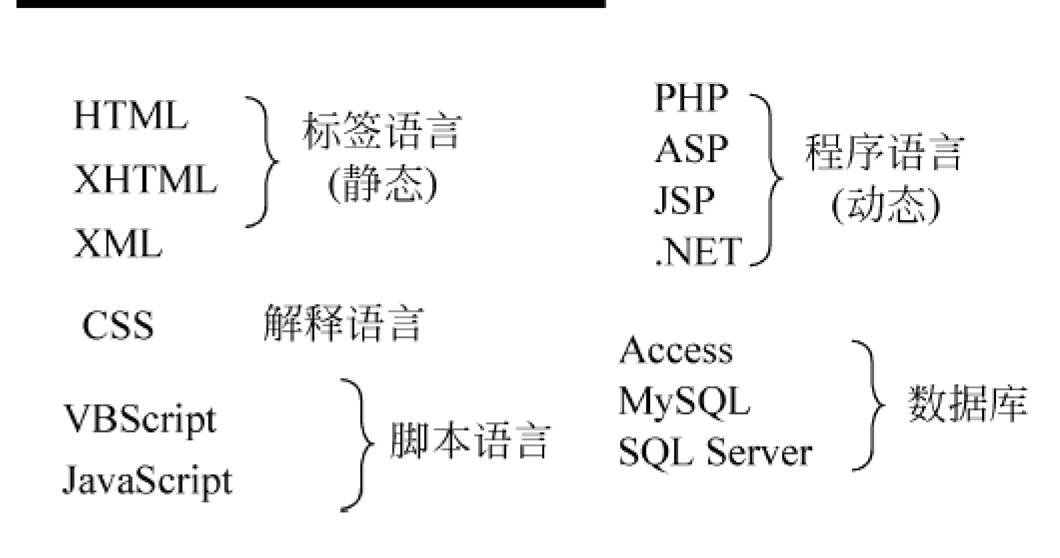


图 1.7　网页相关技术分类

1.4　网站分类与网站托管

1.4.1　网站分类

(1) 门户网站。这类网站是一种综合性网站，涉及领域非常广泛，包含文学、音乐、影视、体育、新闻和娱乐等方方面面的内容，具有论坛、搜索和短信等功能。现在国内较著名的门户网站有搜狐(www.sohu.com)、网易(www.163.com)等。

(2) 个人网站。个人网站具有较强个性化，是以个人名义开发创建的网站，其内容、样式、风格等都是非常有个性的。

(3) 专业网站。这类网站具有很强的专业性，通常只涉及某一个领域，内容专业。如榕树下网站(www.rongshuxia.com)即是一个专业文学网站。

(4) 职能网站。职能网站具有专门的功能，如政府职能网站等。目前逐渐兴起的电子商务网站也属于这类网站，较有名的电子商务网站有京东商城(www.360buy.com)、卓越网(www.joyo.com)和当当网上书店(www.dangdang.com)等。

1.4.2　网站部署方式

在一个网站制作完成后，需要发布出去，让人们看到网站中的信息内容，就要考虑网站部署的问题。目前来说，个人计算机上网一般采用移动、电信或者联通的网络，个人计算机使用这些网络的时候是没有固定 IP 地址的，所以制作好的网页是不能通过个人计算机直接发布出去的。想要发布网站，一般通过以下几种方式之一完成：Web 空间、虚拟主机技术、主机托管技术、独立服务器技术。

(1) Web 空间。购买空间服务商的 Web 空间，存放网站页面内容的计算机存储空间。空间服务商有固定的 IP 地址，网页制作者可以通过 FTP 远程上传网站，发布网站。

(2) 虚拟主机技术。相对于真实主机而言，采用特殊的软、硬件技术把一台完整的服务器主机分成若干个主机。即把真实主机的硬盘分成若干份，租给不同的用户，但每一份都有独立的域名与 IP 地址，具有完整的互联网服务器功能，但共享真实主机的 CPU、RAM、操作系统、应用软件等。

虚拟主机技术的优点如下。

① 缓解互联网上 IP 及服务器等资源的不足。

② 节省投资,如服务器、硬件设备等。

③ 无须租用专线,迅速拥有自己的域名及专属网站。

④ 管理简易,ISP 完成绝大多数管理工作。

⑤ 只需对自己的信息进行远程维护,无须对硬件、操作系统及通信线路进行维护。

⑥ 投资小,建立速度快,安全可靠,无须软硬件配置,无须技术支持。

虚拟主机技术的缺点如下。

① 资料保密性差。

② 只能远程控制。

③ 容易过载,性能下降。

(3) 主机托管技术。指的是用户将自己的独立服务器寄放在互联网服务商的机房,日常系统维护由互联网服务商代理进行,可以节约大量的机房环境、设备维护等资金。

主机托管技术优点如下。

① 可靠性是最基本的因素。

② 重视安全问题。

③ 随时升级。

(4) 独立服务器技术。是指用户的服务器从 Internet 的接入到维护、管理完全由用户自己操作。一般是通过租用专用线路不间断地连接 Internet。要求考虑硬件设施、软件系统、接入方式、防火墙、技术人员配备等。

小结

本章介绍了网页与网站的基本概念,以及网页传送涉及的一些网络规则和概念,例如因特网、HTTP、IP 地址、域名等,与网页与网站相关的概念和一些工具软件,HTML、CSS、JavaScript、PHP 等语言,数据库,浏览器等。在网站制作完成后,提供了一些发布网站空间部署方案。

习题

1. 网页是什么?
2. 网站是什么?
3. IP 地址是什么?
4. 域名是什么?
5. URL 是什么?
6. 网站部署方式有哪些?

第2章 HTML基础

本章要点：

- HTML 概述；
- HTML 标签概述；
- HTML 标签及使用。

WWW(World Wide Web,万维网)是一种建立在 Internet 上全球性的、交互性的信息资源网络。要把信息发布到全球,必须使用一种大多数计算机能够识别的语言。在 WWW 上,通常使用的语言是 HTML。对于网页设计和制作人员,尤其是开发动态网站的编程人员来说,HTML 是进行网页设计和制作必须掌握的基础知识。

2.1 HTML 概述

2.1.1 HTML 的基本概念

HTML 是英文 HyperText Markup Language 的缩写,中文意思是“超文本标记语言”,它是用来描述网页的一种语言。HTML 不是一种编程语言,而是一种标记语言,通过对网页要显示的内容中添加标记符来告诉浏览器如何显示网页。这样,当用户使用 Web 浏览器打开网页时,浏览器读取网页文件(HTML 文档),根据显示内容周围的 HTML 标记符解释和显示各种内容,而 HTML 标记符不会在浏览器中显示。

用 HTML 编写的文档称为 HTML 文档,扩展名是.html 或.htm。HTML 文档也被称为网页,该文档中包含 HTML 标签和纯文本。

2.1.2 一个简单的 HTML 实例

在介绍 HTML 文档基本结构之前,先来看一个简单的 HTML 文档及其在浏览器上的显示结果。下面使用文本编辑器(如 Windows 系统中的记事本软件)编写一个 HTML 文档。HTML 文档名称为“我的第一个网页.html”。

【例 2.1】 简单的 HTML 实例——我的第一个网页.html。

```
<html>
  <head>
     <title>网页标题</title>
```

```
    </head>
    <body>
      网页正文
    </body>
  </html>
```

运行效果如图 2.1 所示。



图 2.1　HTML 示例

2.1.3　HTML 基本结构

要编写 HTML 文档,首先要了解 HTML 文档的基本结构。HTML 文档结构分为文档头和文档体部分。结构如图 2.2 所示。

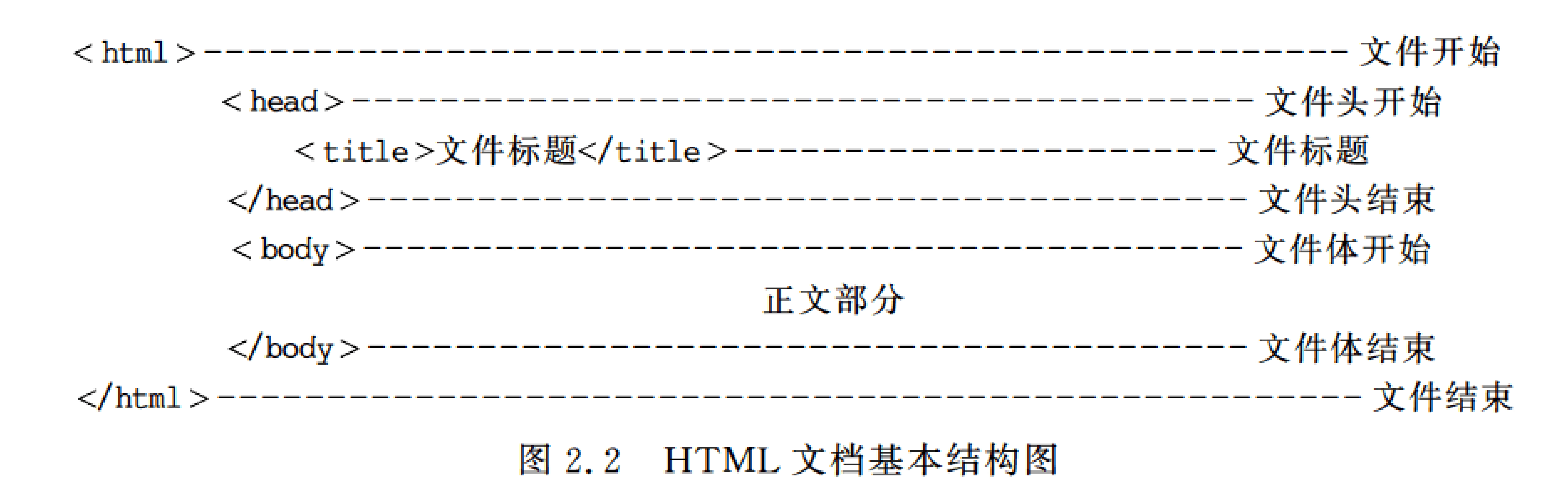


图 2.2　HTML 文档基本结构图

从上述代码中可以看出 HTML 文档的基本结构。最基本的 HTML 文档包括:HTML 标签<html>和</html>、文档头标签<head>和</head>、文档标题标签<title>和</title>、文档体标签<body></body>。注意:HTML 文档标签不区分大小写。

1. <html>标签

在任何一个 HTML 文档中,最先出现的 HTML 标签是<html>,它用于表示该文档是以超文本标记语言(HTML)编写的,这样在解释执行的时候就告诉浏览器或其他阅读该页的程序,此文件为一个 Web 页。<html>标签成对出现。首标签<html>和尾标签</html>

分别位于文件的最前面和最后面，网页的其他所有内容都位于这两个标签之间。该标签不带有任何属性。

2. 文档头标签<head>

文件头标签<head>位于 Web 页的开头，成对出现，即<head>文档头内容</head>。文件头部分不包括 Web 页的任何实际内容，而是提供一些与 Web 页有关的特定信息，如用来规定网页的标题和编码方式等信息。如果 HTML 文件并不需要相关信息时，可省略<head>与</head>标签。

3. 文档标题标签<title>

在浏览器中，网页名称显示在 Web 浏览器窗口的标题栏中，网页名称对浏览器的收藏功能很有用。网页名称写在标签<title>和</title>之间，<title>标签成对出现，即<title>文档标题</title>，且该标签位于<head>与</head>标签之中。

4. 文档体标签<body>

<body>标签成对出现，<body>和</body>包含 Web 页的内容，文本、图形、图像、链接等网页要显示到浏览器窗口中的内容都位于该标签内。文档体标签中的文字，如果没有其他标签修饰，则将以无格式的形式显示。

2.1.4　编写第一个 HTML 文档

HTML 是纯文本类型的语言，可以使用任何文本编辑器(如 Windows 系统中的记事本程序)编写。需要注意的是，要将这些代码保存为网页格式，即在保存时将文件的扩展名改写为.html 或.htm。也可用可视化软件来编写，如 Microsoft 公司的 FrontPage、Adobe 公司的 Dreamweaver 等软件。

下面用记事本编写第一个 HTML 文档。操作步骤如下。

(1) 选择“开始”|“程序”|“附件”|“记事本”命令，打开记事本程序。

(2) 在记事本中输入 2.1.2 节中的实例代码。

(3) 选择菜单中的“文件”|“保存”命令，弹出“另存为”对话框。选择文件保存路径，在“文件名”框中输入文件名称，注意文件的后缀名为.html 或.htm。然后单击“保存”按钮，创建完成。

使用浏览器浏览页面效果，操作步骤：右击刚刚创建的 HTML 文件，在弹出的快捷菜单中选择打开方式，选择 IE 浏览器或其他浏览器，这样可以在浏览器中看到最终的页面效果。

可以使用浏览器查看网页源代码。操作步骤：选择浏览器菜单“查看”|“查看源代码”命令，则在记事本中显示目前所打开页面的源代码。

说明：这里编写的 HTML 网页属于静态网页，该类型网页只给用户提供浏览功能，用户不能和服务器之间进行交互。本书后面章节介绍的 PHP 语言，是为了编制动态网页。

2.2 HTML 标签概述

2.2.1 HTML 标签概念

HTML 标记标签通常被称为 HTML 标签。HTML 标签是由尖括号包围的关键词,如<html>。"<"和">"是任何标签的开始和结束。

使用标签有以下三大作用。

(1) 格式化文本。可以通过使用设定文本颜色、字体、大小以及对齐方式的标签,以及设定文本是标题文本还是段落文本的标签,实现类似于文字处理软件的排版功能。

(2) 进行网页布局。通过使用表格、层等标签实现对网页内容进行布局。

(3) 实现网页超媒体链接。通过使用链接的标签,实现不同页面的跳转以及不同主题的跳转功能。

2.2.2 HTML 标签的分类

HTML 标签根据其格式分为两类:单标签和双标签。单标签只需单独使用就能完整地表达意思。如回车换行标签
。双标签成对出现,如<html>和</html>。大多数 HTML 标签成对出现,标签对中的第一个标签是开始标签,第二个标签是结束标签,结束标签的标签总是在开始的标记前加一斜杠。双标签仅对包含在其中的文件内容部分发生作用。如标题标签<H1>和</H1>之间的文本被显示为标题。

2.2.3 HTML 元素概念及其使用

HTML 元素指的是从开始标签到结束标签的所有代码。例如<p>This is a paragraph</p>,元素内容为 This is a paragraph。即 HTML 元素以开始标签起始,以结束标签终止,元素的内容是开始标签与结束标签之间的内容。没有内容的 HTML 元素被称为空元素。空元素是在开始标签中进行关闭(以开始标签的结束而结束)。例如,
就是没有关闭标签的空元素。在开始标签中添加斜杠,例如
,是关闭空元素的正确方法。在 XHTML、XML 以及未来版本的 HTML 中,所有元素都必须被关闭。

大多数 HTML 元素可拥有属性,属性为 HTML 元素提供附加信息,属性总是以"名称/值"的形式出现,比如:name="value"。属性总是在 HTML 元素的开始标签中规定。例如,<body>标签定义 HTML 文档的主体。<body bgcolor="yellow">拥有关于背景颜色的附加信息。属性名称为"bgcolor",属性值为"yellow"。

大多数 HTML 元素可以嵌套(可以包含其他 HTML 元素)。HTML 文档由嵌套的 HTML 元素构成。

【例 2.2】 HTML 元素概念实例。

```
<html>
   <body>
     <h1>我的第一个网页 </h1>
```

```
        <!-- 文件说明：以下内容中的回车和空格在浏览器显示时会忽略 -->
      <p> 春晓
           春眠不觉晓，
             处处闻啼鸟.
               夜来风雨声，
                花落知多少.
      </p>
    </body>
  </html>
```

上面的例子包括三个 HTML 元素。以下为几点说明。

(1) 任何回车和空格在源代码中不起作用，即浏览器会忽略源代码中的排版(省略多余的空格和换行)。

(2) HTML 源代码中可以添加注释，语法格式以“<!--”开始，以“-->”结束。注释语句只出现在源代码中，不会在浏览器中显示。如代码：

```
<! -- 文件说明：第一个 HTML 文件 -- !>
```

2.3　HTML 标签及使用

2.3.1　HTML 文件基本标签

基本标签已在 2.1.3 节中做过介绍，下面重点介绍一下<head>头部信息标签。

<head>…</head>标签出现在<html>标签后面，用来说明文件的相关信息，如文件的标题、搜索引擎可用的关键字等。标签之间的内容不会在浏览器文档窗口中显示，但是，其间的元素有特殊重要意义。

<head>标签中包含 meta 元素，一般用来定义页面信息的名称、关键字、作者等。在一个 HTML 文档中可以有多个 meta 元素。meta 元素的属性有两种：name 和 http-equiv，其中 name 属性主要用于描述网页，以便于搜索引擎查找；http-equiv 属性用于回应给浏览器一些有用信息，帮助浏览器正确和精确地显示内容。

(1) 常用的 name 属性包括 description、keywords、author 等。

① keywords 设置页面关键字。

用法：<meta name="keywords" content="关键字 1，关键字 2，关键字 3">

作用：设定了“关键字 1”，“关键字 2”，“关键字 3”这三个词语作为该页面的关键字。这样可帮助搜索引擎对该网页进行查找和分类。所有设定的关键字不会在网页中显示。其中 name 为属性名称，本例中设置为 keywords，即设置网页的关键字属性。content 中定义了具体的关键字的内容。

② description 设置页面描述。

用法：<meta name="description" content="关于 HTML 的介绍">

作用：为了便于搜索引擎的查找，设置了网页的描述为“关于 HTML 的介绍”，同关键字一样，设置的页面描述也不会在网页中显示出来。

③ author 设定作者信息。

用法：<meta name="author" content="作者名字">

作用：用于设置页面制作者的姓名及个人信息，这样可在源代码中保留作者希望保留的信息。

(2) 常用的 http-equiv 属性包括 Content-Type、refresh、expires 等。各个属性的作用如下。

① Content-Type 设定网页文字及语言。

用法：<meta http-equiv="Content-Type" content="text/html;charset=gb2312">

作用：设定语言的编码方式为简体中文。

② refresh 设定网页的定时跳转。

用法：<meta http-equiv="refresh" content="3;url=http://www.hao123.com">

作用：从当前页面在经过指定的一段时间之后将自动跳转到指定的网页。refresh 表示网页的刷新，在 content 中设定刷新的时间和刷新后的地址，时间和链接地址之间用分号分隔。默认情况下，跳转时间以 s 为单位。上面的语句表示网页将在 3s 内跳转到 URL 为 http://www.hao123.com 的网页。

③ expires 设定有效期限

用法：<meta http-equiv="expires" content="Wed,14 September 2014 23:00:00 GMT">

作用：设置网页到期时间，一旦超过到期时间必须到服务器上重新调用。到期时间必须是 GMT 时间格式，即星期、日、月、年、时、分、秒，这些时间都使用英文和数字进行设定。上面的语句表示设置了网页的到期时间为 2014 年 9 月 14 日 23:00:00。

2.3.2 标题标签

HTML 中提供了 6 个级别的标题实现以某种固定的字号显示文字，通过<h1>～<h6>等标签显示 1～6 级标题。语法格式如下：

```
1 级标题：<h1>…</h1>
2 级标题：<h2>…</h2>
    …
6 级标题：<h6>…</h6>
```

其中，<h1>定义最大的标题，<h6>定义最小的标题。

【例 2.3】 标题标签实例。

```
<html>
  <head><title>标题示例</title></head>
  <body>
    <h1>1 级标题</h1>
    <h2>2 级标题</h2>
    <h3>3 级标题</h3>
    <h4>4 级标题</h4>
    <h5>5 级标题</h5>
    <h6>6 级标题</h6>
  </body>
</html>
```

网页效果如图 2.3 所示。



图 2.3 标题文字效果

2.3.3 字体标签

网页编辑中，可以直接在文档的主体部分(<body>和</body>标签之间)输入文本，这些文本会显示在页面中，可使用<font>标签设置文本不同的显示效果。该标签有三个常用的属性，face 属性可以设置不同字体效果；size 属性可设置字体的大小(即字号)，其大小是相对于默认字体而言。文字字号可以设置为 1～7，默认文字的大小与 3 号字相同，数值越大，文字也越大；color 属性可设置文字的颜色。关于颜色属性值可以使用不同的方式进行设置，如可以使用颜色名(如 color＝"red")，也可以设置为十六进制的颜色值(color＝"＃FF0000")，或将颜色属性值设置为 RGB(如 color＝rgb(255,0,0))，这三种设置效果均相同。语法格式如下：

```
<font  face = "字体" size = "文字字号" color = "颜色代码">应用该效果的文字</font>
```

【例 2.4】 设置文字字体格式实例。

```
<html>
  <head><title>设置字体格式</title></head>
  <body>
    <font face = "Times New Roman" >本行文本字体为 Times New Roman </font><br/>
    <font size = "6" >六号字体 </font><br/>
    <font>默认字体大小 </font><br/>
    <font face = "黑体" size = "3" color = "red">本行文本字体为黑体,字号为 3 号字,颜色为红
色</font><br/>
    <font color = "blue" >本行文本颜色为蓝色,其他属性为默认设置</font><br/>
    <font color = "ff0000" >本行文本颜色值(红色)用十六进制表示</font><br/>
  </body>
</html>
```

在浏览网页时，可以看到一些特殊效果的文字，如粗体字、斜体字以及带下划线的文字，这些文字效果可以通过设置 HTML 标签来实现。使用方法如下：

```
<strong>粗体文字的效果</strong>
<b>黑体文字的效果</b>
<i>斜体文字的效果</i>
<h>文字加下划线的效果</h>
```

使用<marquee>标签可以使文字在网页中移动，即实现滚动文本效果。使用方法如下：

```
<marquee bgcolor="yellow" width="300" direction="right" scrollmount="10">
    欢迎光临我的网页
</marquee>
```

上面的代码实现了“欢迎光临我的网页”文字从页面左边到右边的移动效果，bgcolor 属性用于设置滚动文字的背景颜色，scrollmount 用于设置文字移动速度。关于移动文字还可以通过设置<marquee>标签的属性值用于实现其他效果。<marquee>标签的属性及说明如表 2.1 所示。

表 2.1 <marquee>标签属性

属　性	说　明	取　值
width	文本区域宽度	px(像素值)
height	文本区域高度	px(像素值)
direction	移动方向	left、right、up、down
scrollmount	移动速度	px(像素值)
loop	循环次数	1,2,…
bgcolor	背景颜色	颜色值
align	对齐方式	top、middle、bottom

2.3.4 格式控制标签

格式控制主要包括段落、换行、水平线和空格。其中，段落通过<p>标签来实现，效果是可以新建一个段落并加上一个空行；如果编辑网页内容只是需要换行，使用
标签，一个
标签代表一个换行，连续的多个
标签可以实现多次换行；<hr>标签用于在页面中添加一条默认样式的水平线。

【例 2.5】 格式控制标签实例。

```
<html>
    <head><title>格式控制标签——段落等</title></head>
    <body>
        <h6>段落等标签使用实例</h6>
        <hr/>
        <p>第一段内容</p>
        <p>第二段内容</p>
        <p>第三段内容<br/>第三段内容第 2 行(仍为第三段内容)<br/>第三段内容第 3 行(仍为
第三段内容)</p>
  </body>
</html>
```

段落居中对齐使用<center>标签。例如，在上面的实例中实现第一段内容居中显示可在<p>标签之前加入<center>标签，</p>之后加入居中对齐尾标签</center>，代码如下：

```
<center>
  <p>段落一内容,该段落居中对齐</p>
</center>
  <p>段落二内容,该段落没有进行段落对齐方式设置</p>
```

另外，段落对齐方式也可以在<p>标签中添加 align 属性实现，align 属性取值可以是下列取值之一：center(居中对齐)、left(左对齐)、right(右对齐)。默认为左对齐方式。

<hr>水平线标签的常用属性包括 width、color、align、size 和 noshade 等，分别用来设置水平线的宽度、颜色、对齐方式、高度和无阴影效果等。align 属性取值可以是下列取值之一：center(居中对齐)、left(左对齐)、right(右对齐)。例如<hr width="555" size="5" align="left" color="blue">，设置水平线对齐方式为页面左对齐。

一般情况下，在网页中输入文字时，如果在段落开始增加了空格，在使用浏览器时看不到这些空格。如果需要保留空格的效果，需通过空格代码来代替。语法格式为： 。

在网页中可以有多个空格，一个" "只代表一个半角空格，多个空格可以多次使用这个符号。

2.3.5　图片标签

HTML 文档能将不同格式、不同路径的图片插入到网页中。HTML 插入图片使用<img>标签，注意，该标签是单标签。该标签常用的属性包括 src、alt、width 和 height 等，分别用来设置图片的路径、替代文字、宽和高等。基本语法格式如下：

```
<img src="URL"  alt="替代文本"/>
```

其中，URL 指存储图像的位置，有绝对路径和相对路径两种表示方式；alt 属性值用来为图像定义一串预备的可替换的文本。在浏览器无法载入图像时，在图片位置显示替代文本，如果图片已显示，则当鼠标指向该图片时显示替代文本。

HTML 中有两种路径写法：绝对路径和相对路径。

绝对路径指文件的完整路径，如"D:\xscj\images\pic1.jpg"是 pic1.jpg 这个文件的绝对路径。

相对路径指的是从当前文件所在位置出发找到其他文件或文件夹的路径关系。在网页制作中经常使用该方法。因为网页都是在本地计算机上制作完成后再上传到 Web 服务器，本地计算机和服务器上的目录结构不同。所以建议在网页中使用到的任何资源，首先复制到网页专用文件夹中，例如图片文件夹(images)，尽量使用相对路径，地址中不要出现本地驱动器盘符，例如"images\pic1.jpg"。

相对路径的引用方式如下。

(1) 被引用文件与当前引用文件出现在同一个目录：直接写被引用的文件名，"*.*"；

(2) 被引用文件出现在当前引用文件所在目录的下一级目录中：直接写下一级目录文件的文件路径，"文件夹名称*.*"。

关于<img>标签的属性及说明如表 2.2 所示。

表 2.2 <img>标签属性说明

属性	属性说明	可用值
width	设置图像宽度	px(像素值)
height	设置图像高度	px(像素值)
align	设置图像悬浮效果	left、right、top、middle、bottom(默认对齐方式)
vspace	设置图像上下空白	px(像素值)
hspace	设置图像左右空白	px(像素值)

【例 2.6】 图片标签实例。

```
<html>
  <body>
   <h2>未设置对齐方式的图像:</h2>
   <p>请注意,如果您把鼠标指针移动到图像上,大多数浏览器会显示文本.</p>
   <img src="imgs/pic1.jpg" alt="图片文本说明" />
   <p>通过改变 img 标签的"height"和"width"属性的值,您可以放大或缩小图像.</p>
   <img src="imgs/pic1.jpg" width="100" height="100"><br/><br/>
   <img src="imgs/pic1.jpg" width="200" height="200">
   <p>图像 <img src="imgs/pic1.jpg" width="200" height="200">文本</p>

   <h2>设置对齐方式的图像:</h2>
   <hr>
   <p><img src="imgs/pic1.jpg" align="bottom" width="200"height="200">底端对齐 </p>
   <p>
  <img src="imgs/pic1.jpg" align="middle" width="200" height="200">垂直居中对齐</p>
   <p><img src="imgs/pic1.jpg" align="top" width="200" height="200">顶端对齐</p>

    <p>请注意,bottom 对齐方式是默认的对齐方式.</p>
    <p><img src  ="imgs/pic1.jpg" align="left" width="200" height="200">
      带有图像的一个段落。图像的 align 属性设置为"left".图像显示在文本的左侧。
    </p>
    <hr/>
    <p><img src="imgs/pic1.jpg" align="right" width="200" height="200">
      带有图像的一个段落。图像的 align 属性设置为"right".图像显示在文本的右侧。
    </p>
   </body>
</html>
```

2.3.6 链接标签

超链接是网页页面中最重要的元素之一,几乎所有的网站页面都用到超链接。超链接可以是一个字、一个词或者一组词,也可以是一幅图像,单击链接可以从一个网页跳转到另一个网页,也可以实现同一网页中内容之间的跳转等功能。基本的链接方式分为两种:外部链接和内部链接。按链接目标的不同,超链接也可以分为页面链接、内部锚记链接、图片链接、E-mail 链接和下载链接。创建超链接的基本语法格式如下:

```
<a href = "资源地址" >链接文本或图片</a>
```

其中，＜a＞标签表示在 HTML 中创建链接，＜a＞和＜/a＞标签之间的文本、图像或其他 HTML 元素被作为超级链接来显示，href 属性规定链接的目标。对同一台计算机上的文件进行链接，可以采用绝对路径或相对路径来表示，一般建议采用相对路径。例如，＜a href＝"2.html"＞第二个网页＜/a＞，当单击文本"第二个网页"时，从当前页面跳转到另一页面 2.html。链接的目标也可以是 URL 地址，例如，＜a href＝"http://www.nuc.edu.cn"＞中北大学＜/a＞，单击"中北大学"这个链接会打开链接网站首页。

＜a＞标签的属性主要包括 target 属性和 name 属性，以下分别介绍。

1. target 属性

target 属性用于设置被链接的目标在何处显示。默认在原窗口中打开链接。下面的代码会在新窗口中打开文档：

```
<a href = "http://www.nuc.edu.cn/" target = "_blank">中北大学</a>
```

target 属性取值及说明如表 2.3 所示。

表 2.3　target 属性取值及说明

target 属性取值	说　　明
_self	在原窗口显示链接页面
_blank	在新窗口显示链接页面
_parent	在上一级窗口中打开
_top	在浏览器的整个窗口中打开，忽略任何框架

2. name 属性

name 属性实现创建文档内的书签的功能，且书签不会显示到页面。该属性值用于规定锚的名称，当使用命名锚时，可以创建直接跳至该命名锚（比如页面中某一部分内容）的链接，这样无须滚动页面来寻找所要寻找的内容。效果如图 2.4 所示。当单击链接"第二部分"时，直接跳转到当前页面中第二部分内容显示，效果如图 2.5 所示。

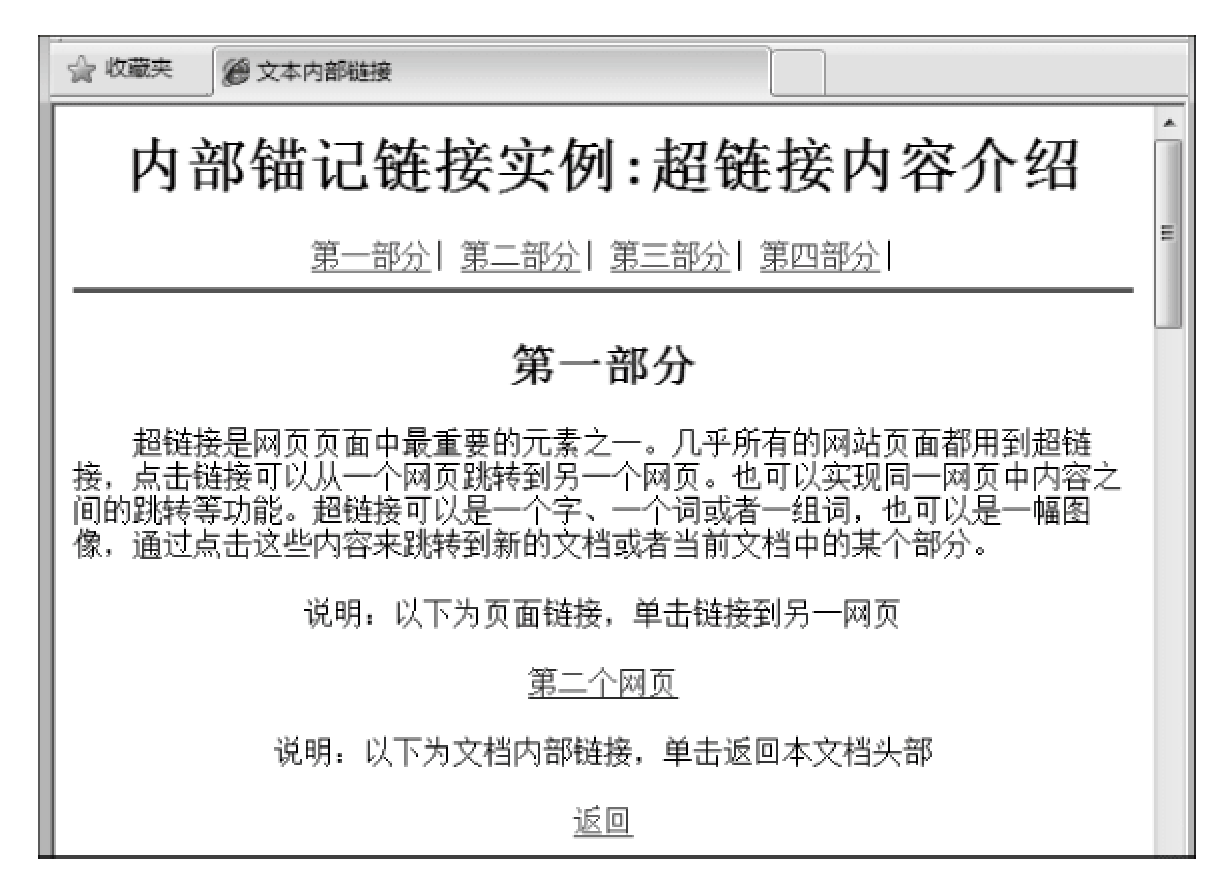


图 2.4　内部锚记链接

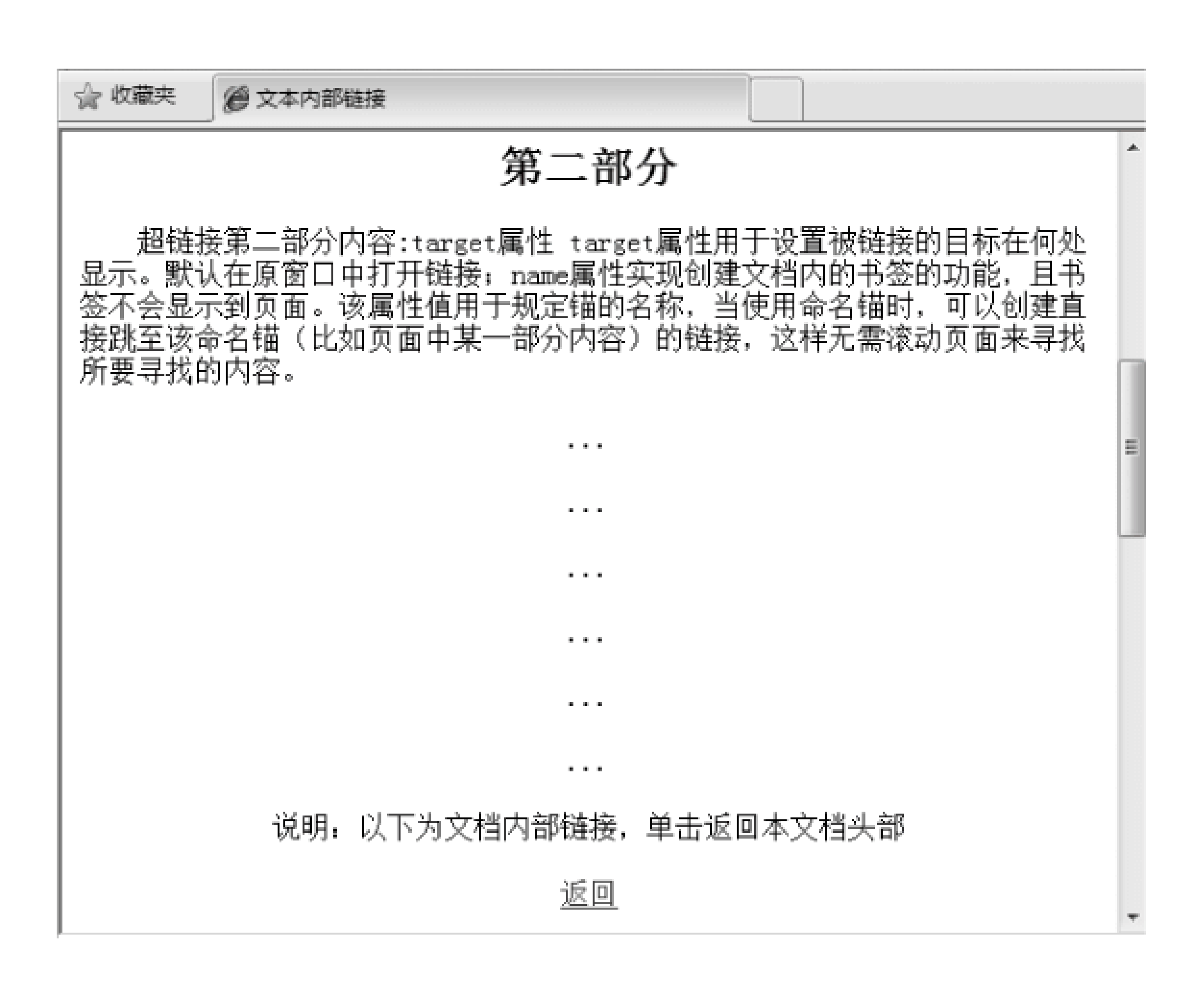


图 2.5 单击链接后的效果

命名锚的语法格式：

```
<a name = "链接文字名称">锚(显示在页面上的文本)</a>
```

下面通过一个实例来学习 name 属性的使用步骤：①在 HTML 文档中对锚进行命名(创建一个书签)：＜a name＝" C1"＞第一部分内容详细介绍＜/a＞；②在同一个文档中创建指向该锚的链接：＜a href＝"＃C1"＞第一部分＜/a＞；也可以在其他页面中创建指向该锚的链接：＜a href＝"2. html ＃C1"＞第一部分＜/a＞。该实例代码将＃符号和锚添加到网页文件 2. html 名称后面，直接单击“第一部分”文字就可以链接到和当前文件在同一文件夹下的 2. html 的 C1 命名锚了。

前面的内容主要涉及页面链接和内部锚记链接，以下介绍其余类型的链接。

创建图片链接的格式：

```
<a href = "资源地址" >< img src = "图形文件地址" ></a>
```

即在＜a＞和＜/a＞之间包含＜img src＝" "＞，通过鼠标单击就可以链接到 href 属性所指定的文件。

创建电子邮件链接的格式：

```
<a href = "mailto:username@163.com " >请联系我们 Email </a>
```

其中，username@163. com 为用户邮箱地址，鼠标单击后直接调用用户计算机上默认邮件收发工具 Outlook 等。

如果希望制作下载文件的链接，只需在链接地址处输入文件所在的地址。下载链接的格式：

```
<a href = "文件所在地址" >链接文字</a>
```

注意：在文件所在地址中设置文件的路径，可以是相对路径，也可以是绝对路径。

空链接：所谓空链接是指在单击链接后，仍然停留在当前页面。在代码中可以通过＃

符号来实现空链接。语法格式：

```
<a href="#">链接文字</a>
```

【例 2.7】 超链接综合实例代码。实现效果如图 2.4 所示。

```
<html>
  <head><title>文本内部链接</title></head>
  <body>
  <h1 align=center><a name="C0">内部锚记链接实例:超链接内容介绍</a></h1>
  <center>
     <a href="#C1">第一部分</a>|
     <a href="#C2">第二部分</a>|
     <a href="#C3">第三部分</a>|
     <a href="#C4">第四部分</a>|
  </center>
  <hr size=3 color=red align=left>
  <center><h2><a name="C1">第一部分</a></h2></center>
  <p>    超链接是网页页面中最重要的元素之一。几乎所有的网站页面都
用到超链接,单击链接可以从一个网页跳转到另一个网页。也可以实现同一网页中内容之间的跳转
等功能。超链接可以是一个字、一个词或者一组词,也可以是一幅图像,通过单击这些内容来跳转到
新的文档或者当前文档中的某个部分。</p>
   <center><p>说明: 以下为文档内部链接,单击返回本文档头部</p></center>
   <center><a href="#C0">返回</a></center>

   <center><h2><a name="C2">第二部分</a></h2></center>
   <p>    超链接第二部分内容:target 属性 target 属性用于设置被链接的
目标在何处显示。默认在原窗口中打开链接; name 属性实现创建文档内的书签的功能,且书签不会
显示到页面。该属性值用于规定锚的名称,当使用命名锚时,可以创建直接跳至该命名锚(比如页面
中某一部分内容)的链接,这样无须滚动页面来寻找所要寻找的内容。
   </p>
   <center><p>说明: 以下为文档内部链接,单击返回本文档头部</p></center>
   <center><a href="#C0">返回</a></center>

   <center><h2><a name="C3">第三部分</a></h2></center>
   <p>    超链接第三部分内容:基本的链接方式分为两种: 外部链接和内
部链接。按链接目标的不同,超链接也可以分为页面链接、图片链接、E-mail 链接、下载链接和内部
锚记链接。前面的内容主要涉及页面链接和内部锚记链接,以下介绍其余类型的链接。</p>
   <center><a href="#C0">返回</a></center>

    <center><h2><a name="C4">第四部分</a></h2></center>
    <h1>超链接第四部分内容:总结</h1><hr/>
    页面链接(链接到): <a href="http://www.nuc.edu.cn" target="_blank">中北大学</a><br/>
    <p>说明: 以下为文档内部链接,单击返回本文档头部</p>
    内部锚记链接: <a href="#C0">返回</a><br/>
    图片链接:
   <a href="2.html"><img src="imgs/pic1.jpg" width="80" height="80"></a><br/>
    E-mail 链接: <a href="mailto:username@163.com" >请联系我们 Email</a><br/>
    下载链接: <a href="压缩文件.zip">链接文字</a></p>
    <center><a href="#C0">返回</a></center>
</body>
</html>
```

2.3.7 表格标签

使用表格可以清晰地显示数据，在 HTML 中还可以使用表格来实现网页布局功能。创建表格使用的基本标签如下。

(1) <table>…</table>　　定义表格

(2) <caption>…</caption>　　定义表格标题

(3) <tr>… </tr>　　定义表格中的一行，如果是多行，则多次出现

(4) <th>…</th>　　定义表头单元格，文本黑体居中

(5) <td>…</td>　　定义普通单元格，每行被分为若干个单元格

【例 2.8】 表格实例。

```
<html>
  <head><title>表格实例</title></head>
  <body>
    <table border = 3>
      <caption>XXX 班学生成绩表</caption>
       <tr>        <! -- 表格第一行 -->
         <th>学号</th><th>姓名</th><th>语文</th><th>数学</th><th>平均分数</th>
       </tr>
       <tr>        <! -- 表格第二行 -->
          <td>0001</td><td>张三</td><td>92</td><td>98.5</td><td>95.3</td>
       </tr>
       <tr>        <! -- 表格第三行 -->
          <td>0002</td><td>李四</td><td>88</td><td>82</td><td>85</td>
       </tr>
       <tr>        <! -- 表格第四行 -->
          <td>0003</td><td>王五</td><td>68.5</td><td>90</td><td>79.3</td>
       </tr>
    </table>
  </body>
</html>
```

该实例创建了一个 4 行 5 列的表格。实例效果如图 2.6 所示。

XXX班学生成绩表

学号	姓名	语文	数学	平均分数
0001	张三	92	98.5	95.3
0002	李四	88	82	85
0003	王五	68.5	90	79.3

图 2.6　表格示例

表格的属性设置如下。

(1) 表格大小尺寸设置：<table width="x" height="y">

说明：x、y 代表具体的像素值。以下同。例如：<table width="400" height="300">。

(2) 边框尺寸设置：<table border="z">

说明：如果不定义边框属性，表格将不显示边框。通过此属性建立有框线或无框线的表格，上例中使用边框属性来显示一个带有边框的表格：<table border="3">。

(3) 单元格边距及单元格间距属性的设置：

```
<table width = "x" border = "y" cellspacing = "z" cellpadding = "z">
```

说明：cellspacing 属性值用来设置单元格的间距，cellpadding 属性值用来设置单元格内容距离单元格边框的像素值。

(4) 单元格中内容对齐方式设置。

两种方式：①左右排列，使用 align 属性。<tr align="#">、<th align="#">、<td align="#">，其中 align 取值可以是 left、center 或 right，分别代表左对齐、居中对齐或右对齐。②上下排列：使用 valign 属性。<tr valign="#">、<th valign="#">、<td valign="#">，其中 valign 取值可以是 top、middle 或 bottom，分别代表顶端对齐、居中对齐或底部对齐。

(5) 表格在网页中的对齐方式设置：<table align="#">。

说明：其中 align 取值可以是 left、center 或 right。

(6) 表格背景颜色或背景图像设置。

表格背景颜色：<table bgcolor="#">。

表格背景图像：<table background="imgs/pic1.jpg">。其中，background 属性值为背景图像文件路径。

行的背景颜色：<tr bgcolor="#">。

单元格背景颜色：<th bgcolor="#">或<td bgcolor="#">。

单元格背景图像：<th background=" imgs/pic1.jpg">或<td background="imgs/pic1.jpg">。

边框颜色：<tr bordercolor="#">。

(7) 跨行或跨列的表格单元格设置。

合并列：<th colspan="#">、<td colspan="#">。

合并行：<th rawspan="#">、<td rawspan="#">。

【例 2.9】 合并单元格实例，效果如图 2.7 所示。

实例核心代码如下。

学号	个人信息			入学信息	
	姓名	性别	年龄	班级	入学年月
001	张三	男	19	060141	2012
002	李四	男	20	060141	2012

图 2.7 合并单元格示例

```
<tr>    <! - 第一行 -->
    <th rowspan="2">学号</th>
    <th colspan="3">个人信息</th>
    <th colspan="2">入学信息</th>
</tr>
<tr>    <! - 第二行 -->
    <th>姓名</th><th>性别</th><th>年龄</th><th>班级</th><th>入学年月</th>
</tr>
<tr>    <! - 第三行 -->
    <td>001</td><td>张三</td><td>男</td><td>19</td><td>060141</td><td>2012
</td>
</tr>
<tr>    <! - 第四行 -->
    <td>002</td><td>李四</td><td>男</td><td>20</td><td>060141</td><td>2012
</td>
</tr>
```

该实例在表头部分填写“学号”内容的单元格跨越三行，填写“个人信息”内容的单元格跨越三列。

网页布局对改善网站的外观非常重要。表格是创建布局的一种简单的方式。大多数网站会把内容安排到多个列中,可以使用<table>标签来创建多列。

【例 2.10】 本例使用三行两列的表格——第一行和最后一行使用 colspan 属性来横跨两列。页面效果如图 2.8 所示。

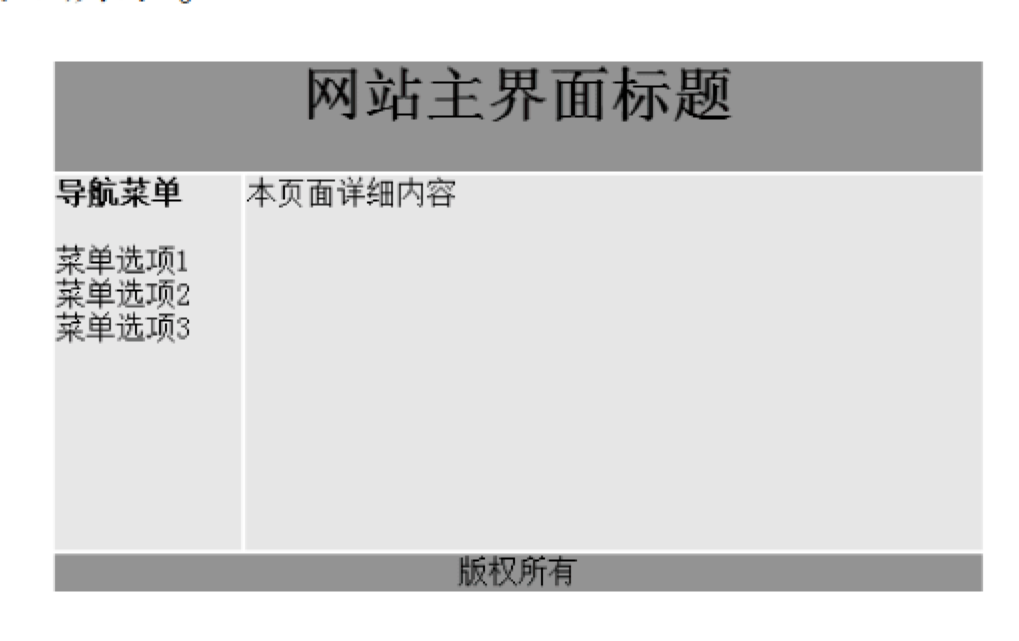


图 2.8　HTML 布局——使用表格

实例核心代码如下。

```
<table width="500" border="0">
  <tr>
      <td colspan="2" style="background-color:#99bbbb">
          <center><h1>网站主界面标题</h1></center>
      </td>
  </tr>
  <tr valign="top">
      <td style="background-color:#ffff99;width:100px;text-align:top;">
           <b>导航菜单</b><br/><br/>
        菜单选项 1<br/>
        菜单选项 2<br/>
        菜单选项 3
     </td>
     <td style="background-color:#EEEEEE;height:200px;width:400px;text-align:top;">
本页面详细内容</td>
   </tr>
   <tr>
      <td colspan="2" style="background-color:#99bbbb;text-align:center;">
版权所有</td>
   </tr>
</table>
```

说明:即使可以使用 HTML 表格来创建漂亮的布局,但设计表格的目的是呈现表格化数据。表格是创建布局的一种简单的方式。在 HTML 中,可以使用其他方式对网页进行布局,如使用<div>元素。

2.3.8　框架标签

前面所介绍的网页都是在浏览器窗口中显示一个网页页面,本节介绍框架标签,通过使用框架,可以在同一个浏览器窗口中显示多个页面。框架的实例效果如图 2.9 和图 2.10

所示。

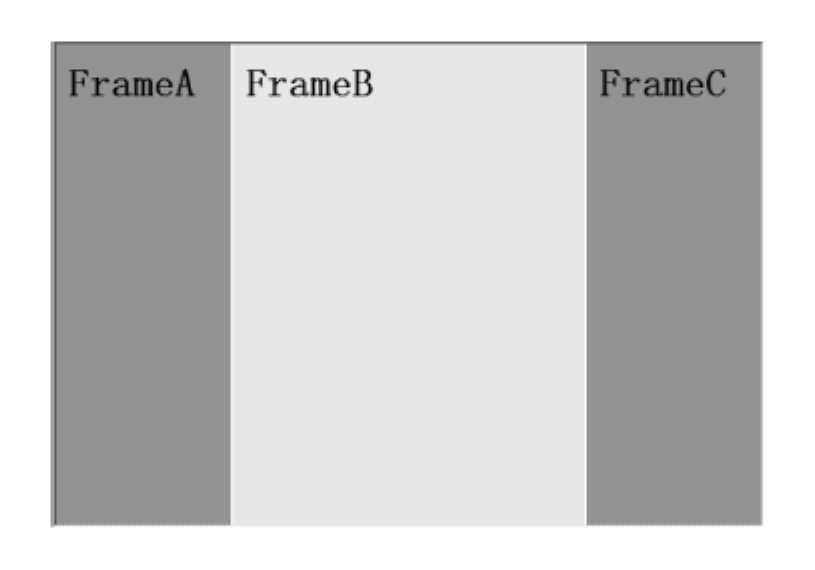



图 2.9　垂直框架

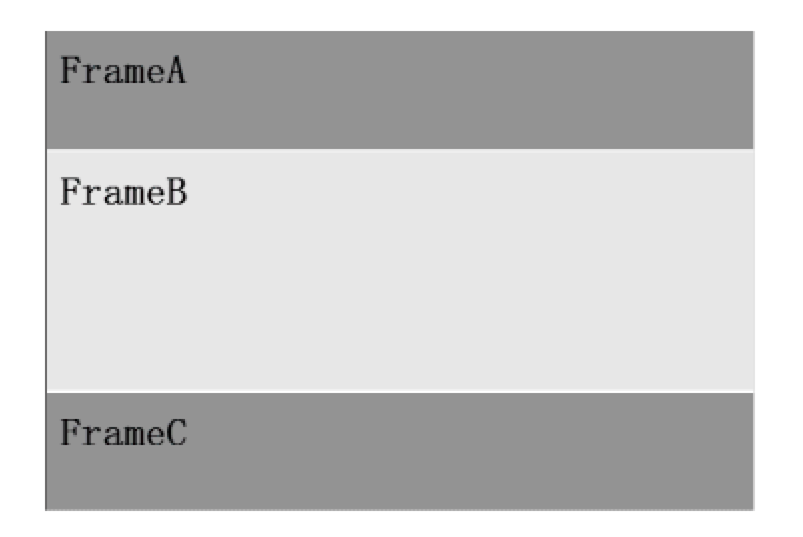


图 2.10　水平框架

垂直框架将浏览器垂直分隔为三部分，每一部分使用不同的文档。水平框架将浏览器水平分隔为三部分，同样，每一部分使用不同的文档，即一个浏览器窗口同时显示三个页面。每份 HTML 文档称为一个框架，并且每个框架都独立于其他的框架。框架主要包括两个部分，一个是框架结构<frameset>，另一个是框架<frame>，其中框架结构标签(<frameset>)定义如何将窗口分割为框架，<frame>标签包含在<frameset>标签之内，该标签定义了放置在每个框架中的 HTML 文档。

1. <frameset>标签属性

每个 frameset 定义了一系列行或列，rows 和 columns 属性值规定了每行或每列占据屏幕的大小。如果使用 rows 属性，用于水平分隔窗口，各个框架窗口的宽度(水平分隔时)或高度(垂直分隔时)值中间用“,”分隔，框架宽度单位可以是绝对像素值、百分数或 * 号。

说明：当符号 * 只出现一次，其中 * 号表示其他子窗口的大小都有明确的宽度或高度值时，该子窗口的大小将根据浏览器窗口的大小而自动调整为剩余部分。当符号 * 出现一次以上时，表示按比例分割浏览器窗口的剩余空间。例如<frameset cols="40%, *, *">，该例将浏览器窗口分割为三列，第一个子窗口在第一列，窗口宽度为整个浏览器窗口宽度的 40%；剩下的空间平均分配给另外两个子窗口。

2. <frame>标签

放在<frameset></frameset>之间，用来标识子窗口，在<frameset>中定义了多少个子窗口，就要有多少个<frame>标签与之匹配，依次定义各个窗口的性质。<frame>标签具有 src 和 name 属性，这两个属性都是必须赋值的。

src 属性用于设置子窗口的源 HTML 文件名，浏览器将会在此窗口中显示 src 指定的 HTML 文件；下面的这个例子中，创建了一个两列的框架集。第一列被设置为占据浏览器窗口的 25%，第二列被设置为占据浏览器窗口的 75%。HTML 文档 frame_a.htm 被置于第一个列中，而 HTML 文档 frame_b.htm 被置于第二个列中，代码如下：

```
<frameset cols = "25 % , * ">
   <frame src = "frame_a.html" name = "Left">
   <frame src = "frame_b.html" name = "Right">
</frameset>
```

name 属性用于设置子窗口的名字，上面的代码将框架集中的第一列，也即左边子窗口命名为“Left”，将框架集中的第二列，也即右边子窗口命名为“Right”，通过该设置，超文本链接标签＜a href＝" " target＝" "＞中的 target 属性就可指定被链接的 HTML 文件将显示在哪一个子窗口中。

【例 2.11】 使用框架导航跳转至指定的页面，本例演示两个框架。左侧的导航框架包含一个链接列表，这些链接将第二个框架作为目标。第二个框架显示被链接的文档。导航框架其中的链接指向目标文件中指定的节。实例效果如图 2.11 所示。

图 2.11 使用框架导航

核心代码如下。

(1) 首页.html。

```
<html>
    <head><title>框架标签的综合示例</title></head>
    <frameset cols="25%,*">
        <frame src="menu.html" scrolling="no" name="Left">
         <frame src="frame_a.html" scrolling="auto" name="Main">
    </frameset>
</html>
```

(2) 左侧的导航框架链接页面 menu.html。

```
<html>
     <head>
     <title>目录</title>
     </head>
     <body>
       <p><font color="#FF0000">目录(请单击链接)</font></p>
       <p><a href="frame_a.html" target="Main">frame_a 页面链接</a></p>
       <p><a href="frame_b.html" target="Main">frame_b 页面链接</a></p>
       <p><a href="frame_c.html" target="Main">frame_c 页面链接</a></p>
     </body>
</html>
```

3. <frame>其他相关属性设置

<frame>其他相关属性设置说明如表 2.4 所示。

表 2.4　<frame>其他相关属性设置

属　性	说　明	取　值
frameborder	设置框架边框	0 或 1,0 为无边框,1 为显示边框
scrolling	设置框架滚动条是否显示	yes,no,auto
noresize	设置是否可调整框架尺寸	noresize 禁止调整,默认为可调整

在 HTML 文档中的一个区域插入一个 HTML 文档,称为浮动框架,也称内联框架,编写代码时不需要在单独的网页中创建框架集,使用<iframe>标签即可建立,建立同时可设置浮动框架的宽度和高度等属性。

基本语法格式如下:

```
< iframe src = "文件地址" height = "值" width = "值" name = "框架名称" scrolling = "值"
frameborder = "值">
```

<iframe>标签的部分属性详细说明如表 2.5 所示。

表 2.5　<iframe>标签的部分属性说明及取值

属　性	说　明	取　值
src	设置浮动框架文件的文件名或其他超链接的网址	
width、height	分别设置浮动框架的宽度、高度	数值
name	设置浮动框架的名称	
frameborder	设置框架边框	0 或 1,0 为无边框,1 为显示边框
scrolling	设置框架滚动条是否显示	yes,no,auto(根据窗口内容决定是否显示)

【例 2.12】　内联框架实例效果如图 2.12 所示。

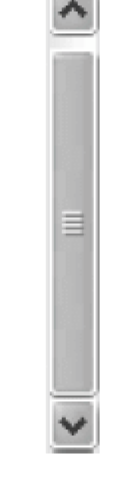



图 2.12　内联框架效果

核心代码如下。

```
<body>
    <iframe src = "/i/eg_landscape.jpg"></iframe>
</body>
```

2.3.9 列表标签

列表是一种以结构化、易读性的方式提供信息的方法。HTML 常用的列表形式包括有序列表和无序列表。

1. 无序列表

无序列表是指列表中列表项的前导符号没有次序，此列项目使用粗体圆点(典型的小黑圆圈)、方框等特殊符号进行标记。无序列表开始于<ul>标签，每个列表项开始于<li>。例如：

```
<ul>
   <li>Coffee</li>
   <li>Milk</li>
</ul>
```

该实例在浏览器中显示如下：

- Coffee
- Milk

设置列表项目前导符号可通过<ul>标签的 type 属性。属性取值为 disc、circle 或 square，分别代表实心圆●、空心圆○、方框□。

列表项内部可以使用段落、图片、链接以及其他列表等。

2. 有序列表

有序列表的每个列表项目使用数字等进行标记，且是有顺序的。有序列表开始于<ol>标签。每个列表项开始于<li>标签。例如：

```
<ol>
  <li>Coffee</li>
  <li>Milk</li>
</ol>
```

该实例在浏览器中显示如下：

1. Coffee
2. Milk

有序列表的列表项除默认使用阿拉伯数字进行标记外，还有其他的排序方式，可通过<ol>标签的 type 属性设置。属性取值：1(阿拉伯数字序列)、a(小写英文字母序列)、A(大写英文字母序列)、i(小写罗马数字序列 i，ii，iii，…)、I(大写罗马数字序列 I，II，III，…)。

3. 列表嵌套使用

【例 2.13】 列表综合应用实例，效果如图 2.13 所示。

实例核心代码如下。

```
<ul type = square>
   <li>第一项</li>
   <li>第二项</li>
   <li>第三项</li>
   <li>第四项</li>
</ul>

<ol start = 5>
  <li type = a>第一项</li>
  <li>第二项</li>
  <li>第三项</li>
  <li>第四项</li>
</ol>

<ul>
<li>直辖市
  <ol>
     <li>北京市
     <li>天津市
     <li>上海市
     <li>重庆市
  </ol>
</li>
<li>省
   <ol>
     <li>福建省
     <li>广东省
     <li>湖北省
     <li>...
   </ol>
</li>
</ul>
```

图 2.13 列表示例

2.3.10 表单标签

表单(Form)是 HTML 的一个重要组成部分，用来收集浏览者的信息，并将信息提交给服务器进行处理。表单中包含很多表单控件，常用的表单控件包括单行文本输入框、密码框、多行文本输入框、单选按钮、复选框、下拉列表框、提交和重置按钮。下面分别讲述这些表单及表单控件的标签以及代码格式。

1. 表单标签的格式

<form>是表单的标签，所有的表单控件在使用时必须放在<form>和</form>之间。代码格式：

```
<form action = "url" name = " " method = " "></form>
```

属性解释：

(1) action="url"：定义当提交表单时向何处发送表单数据。

(2) name：定义表单的名称。

(3) method：取值可以是 get 或 post，规定用于发送 form-data 的 HTTP 方法。

2. 文本输入框

当用户要在表单中输入字母、数字等内容时，会用到文本框。代码格式：

```
<input  type="text" name=" " size= " " maxlength= " " value= " ">
```

属性解释：

(1) type="text"：定义单行文本输入框。

(2) name：定义文本框的名称。

(3) size：文本框的宽度，单位为单个字符的宽度。

(4) maxlength：定义最多输入的字符数。

(5) value：定义文本框的初始值。

3. 密码框

一种特殊的文本框，用于输入密码。当输入文字时，文字会被星号或其他符号代替，而输入的文字会被隐藏。代码格式：

```
<input type="password" name=" " size=" " maxlength=" ">
```

属性解释：

(1) type="password "：定义密码框。

(2) name：定义密码框的名称。

(3) size：定义密码框的宽度，单位为单个字符的宽度。

(4) maxlength：定义最多输入的字符数。

4. 多行文本输入框

用户可以输入多行文本。代码格式：

```
<textarea name=" " cols=" " rows=" " wrap=" "></textarea>
```

属性解释：

(1) name：定义多行文本输入框的名称。

(2) cols、rows：分别定义多行文本框的宽度和高度，单位为单个字符的宽度。

(3) wrap：定义输入内容大于文本框时显示的方式。

5. 隐藏域

隐藏域用来收集或发送信息的不可见元素，对于网页的访问者来说，隐藏域是看不见的。当表单提交时，隐藏域会将设置的信息发送到服务器上。代码格式：

```
<input type="hidden" name=" " value=" ">
```

属性解释：

(1) type="hidden"：定义隐藏域。
(2) name：定义隐藏域的名称。
(3) value：定义隐藏域的值。

6. 单选按钮

当需要访问者在待选项中选择唯一的答案时，需要用到单选框。代码格式：

```
<input type="radio" name=" " value=" ">
```

属性解释：
(1) type="radio"：定义单选按钮。
(2) name：定义单选按钮的名称。
(3) value：定义单选按钮的值。

7. 复选框

复选框允许在待选项中选中一个以上的选项。每个复选框都是一个独立的元素，都必须有一个唯一的名称。代码格式：

```
<input type="checkbox" name=" " value=" ">
```

属性解释：
(1) type="checkbox"：定义复选框。
(2) name：定义复选框名称。
(3) value：定义复选框的值。

8. 下拉列表框

下拉选择框允许在一个有限的空间设置多种选项。代码格式：

```
<select name=" " size=" " multiple>
  <option value="" selected>…</option>
      …
</select>
```

属性解释：
(1) size 属性：定义下拉列表框的行数。
(2) name 属性：定义下拉列表框的名称。
(3) multiple 属性：表示可以多选，如果不设置本属性，那么只能单选。
(4) value 属性：定义选择项的值。
(5) selected 属性：表示默认已经选择本选项。

9. 文本上传框

文件上传框看上去和其他文本框差不多，只是它还包含一个“浏览”按钮。访问者可以通过输入需要上传的文件的路径或者单击“浏览”按钮选择需要上传的文件。代码格式：

```
<input  type="file"  name=" " size="  " maxlength=" ">
```

属性解释：

（1）type="file"：定义文本上传框。

（2）其他属性同文本框。

10. 提交和重置按钮

1）提交按钮

代码格式：

```
<input type="submit" name="..." value="... ">
```

属性解释：

（1）type= "submit"：定义提交按钮。

（2）name 属性：定义按钮的名称。

（3）value 属性：定义按钮的显示文字。

2）重置按钮

重置按钮用来重置表单。代码格式：

```
<input type="reset" name="..." value="...">
```

属性解释：

（1）type="reset"：定义复位按钮。

（2）name 属性：定义按钮的名称。

（3）value 属性：定义按钮的显示文字。

【例 2.14】 注册表单实例。

```
<body>
   <h1>用户注册 -- 请填写以下信息</h1>
   <form name="form1" method="post" action=" ">
     <table width="500" align="left" bgcolor="#83c4f6">
       <tr>
           <td width="120" height="30">用户名</td>
           <td><input type="text" name="name" ></td>
       </tr>
       <tr>
           <td height="30" >用户密码</td>
           <td><input type="password" name="password"></td>
       </tr>
       <tr>
           <td height="30" >性别</td>
           <td><input type="radio" name="sex" value="男" checked="true">男
               <input type="radio" name="sex" value="女">女</td>
       </tr>
       <tr>
         <td height="30">最高学历</td>
         <td><select name="select">
```

```
                <option>请选择您的最高学历</option>
                <option>高中</option>
                <option>大专</option>
                <option>本科</option>
               <option>研究生</option>
            </select></td>
        </tr>
        <tr>
            <td height="30">最感兴趣的商品</td>
            <td>
    <input type="checkbox" name="inst" value="数码产品">数码产品
    <input type="checkbox" name=" inst " value="服装/服饰时尚">服装/服饰时尚<br>
    <input type="checkbox" name="inst" value="书籍/音像/游戏">书籍/音像/游戏
    <input type="checkbox" name="inst" value="家居/家电/礼品">家居/家电/礼品
<br>
    <input type="checkbox" name="inst" value="珠宝/手表/眼镜">珠宝/手表/眼镜
          </td>
    </tr>
    <tr>
      <td height="30">自我介绍</td>
      <td><textarea name="intro" rows="5" cols="45"></textarea></td>
    </tr>
    <tr>
      <td height="40" colspan="2" align="center" >
         <input type="submit" name="button1" value="注册" />  
         <input type="reset" name="button2"value="重置" />
      </td>
    </tr>
  </table>
 </form>
</body>
```

实例效果如图 2.14 所示。

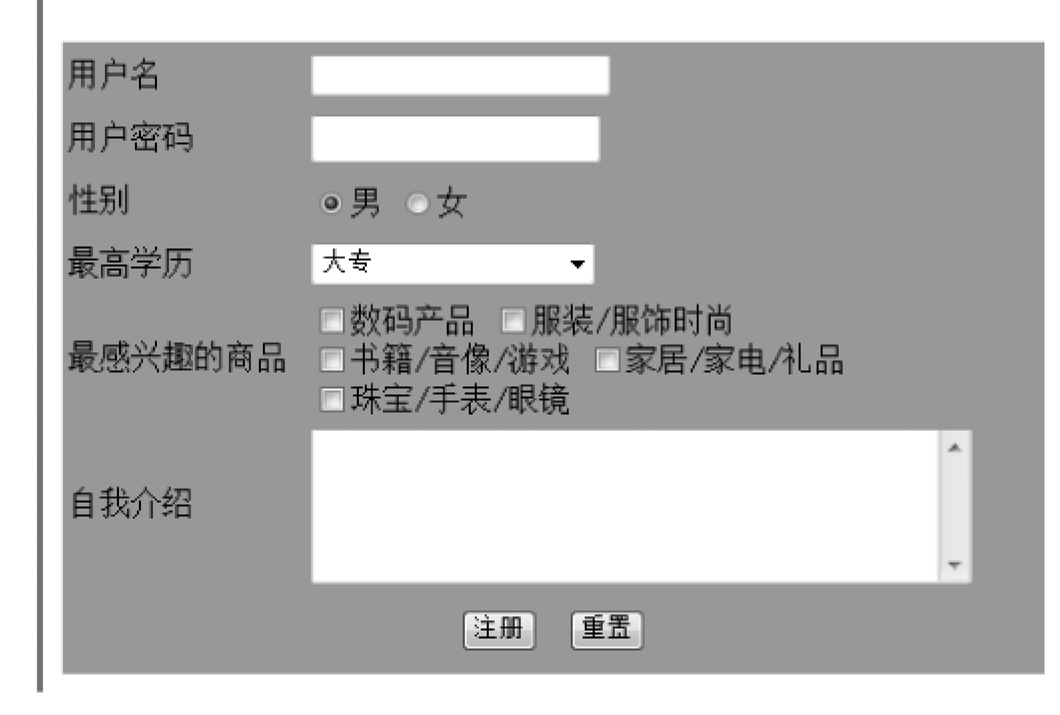


图 2.14 表单示例

2.3.11 多媒体标签

在网页中应用的多媒体主要包括音频、视频和 Flash 等。对于多媒体元素的应用,因为不同的浏览器对多媒体元素标签的支持和处理过程有所不同,所以要考虑在多种浏览器中的显示效果可能不同。

1. 背景音乐

HTML 中插入音频文件常用的格式有:MP3、WAV、MIDI 等。<bgsound>标签可以将声音在后台作为背景音乐播放,使用实例:<bgsound src="bgsound.mp3" loop=-1/>,其中,loop=-1 表示无限循环,也可设置具体值,即声音在循环播放几次后停止。注意:Firefox 和 Opera 浏览器中不支持该标签。

<embed>标签在 IE、Firefox 和 Opera 浏览器中都可以正常应用。这是一个 HTML5 标签,在 HTML4 中是非法的,但是所有浏览器中都有效。

<embed>标签使用实例:

```
<embed height="100" width="100" src="song.mp3" autostart="true" loop="true"/>
```

通过<embed>标签嵌入声音后,网页中会出现播放音频的声音控制器。用户可控制声音的播放和停止。

2. 动画和视频

在网页中创建视频和音频的方法基本相同,通过<embed>标签可以在网页中播放视频或动画文件。常用的视频格式有 MPG、WAV、RM 和 AVI。Flash 动画文件的格式为.swf。下面的 HTML 代码显示嵌入网页的 Flash 视频:

```
<embed src="movie.swf" height="200" width="200"/>
```

另外一种方法:可以使用<object>标签。

<object>标签的作用是在 HTML 页面中嵌入多媒体元素。下面的 HTML 片段显示嵌入网页的一段 Flash 视频:

```
<object data="movie.swf" height="200" width="200"/>
```

下面的代码片段能够显示嵌入网页中的 MP3 文件:

```
<object height="100" width="100" data="song.mp3"></object>
```

2.3.12 块标签<div>与<span>

<div>与<span>标记本身没有具体的显示效果,由其 style 属性或 CSS 来定义,不过两者在使用方法上存在很大的区别。

div 全称 division,意为“区分”,是用来为 HTML 文档内大块的内容提供结构和背景的标签。div 的起始标签和结束标签之间的所包含内容的特性由<div>标签的属性来控制,或者是通过使用样式表来格式化 div 的起始标签和结束标签之间的所有内容。div 标签使

用格式：

```
<div id = "value" align =  "value" class = "value" style = "value">
```

属性解释：

（1）id：为该<div>标签命名，常与 CSS 样式结合，实现对网页元素的控制。

（2）align：用于控制<div>标签中的元素的对齐方式，其值可以是 left、center、right。

（3）class：用于设置<div>标签中的元素的样式。其值为 CSS 样式中的 class 选择符。

先定义一个 CSS 规则：.font{font-size:larger}，该规则定义字体样式。下面的内容多次使用到同一个样式表“.font1”。

```
<h1 class = ".font1 "> H1 标签使用 div 样式</h1 >
<div class = ".font1 "> DIV 标签使用 div 样式</div >
```

（4）style：用于设置<div>标签中的元素的样式。其值为 CSS 属性值，各属性值应用分号分隔。

```
<div style = "width:100 % ;height:300px;font_size:12pt ">使用 div 样式</div >
```

<span>标签和<div>标签非常类似。可以作为插入 CSS 这类风格的容器。但 div 是一块级元素，可以包含段落、标题、表格等。而 span 标签是行内元素，span 的前后是不会换行的，它没有结构意义，纯粹是应用样式。从功能角度来说，<div>标签一般用来作布局，而<span>标签用来制作文字的效果，尤其是标题和链接的效果，例如：

```
<h1 class = " ">< span >…</span ></h1 >
```

<div>…</div>是块，…是行，块里可以含行，即 div 块里可以有 span 行，</div>显示时（分块结束处），系统自动换行，而不换行。

【例 2.15】　<div>与<span>标签实例。

```
<html >
  < body >
    < p > div 标记不同行: </p >
    <div >< img src = " img/pic1.jpg" width = "60" height = "40"></div >
   <div >< img src = " img/pic1.jpg" width = "60" height = "40"></div >
   <div >< img src = " img/pic1.jpg" width = "60" height = "40"></div >
   < p > span 标记同一行: </p >
   < span >< img src = " img/pic1.jpg" width = "60" height = "40">
</span >
   < span >< img src = " img/pic1.jpg" width = "60" height = "40">
</span >
   < span >< img src = " img/pic1.jpg" width = "60" height = "40">
</span >
 </body >
</html >
```

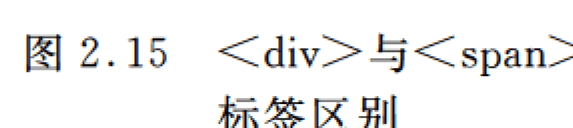


图 2.15　<div>与<span>标签区别

该实例在浏览器中的显示效果如图 2.15 所示。

<div>包围的三幅图像被自动换行分在不同的三行中,而<span>标记的三幅图像没有换行。

2.3.13 样式标签

CSS 是 Cascading Style Sheets 的简称,中文翻译为“级联样式表”,也有人翻译为“样式表”。它是以 HTML 为基础的语言,用于定义网页中内容的格式。其主要特点是可以独立地为网页中的各种对象定义格式,并可将一种格式用于多个网页。

CSS 样式可以将许多网页的格式同时更新,即将站点上所有的网页都使用一个 CSS 文件进行控制,只要修改这个 CSS 文件中相应的代码,那么整个站点的所有页面都会随之发生变动;可以轻松地控制页面的布局。

当浏览器读到一个样式表,它就会按照这个样式表来对文档进行格式化。有以下三种方式来插入样式表:内联样式、内部样式表、外部样式表。

1. 内联样式

当特殊的样式需要应用到个别元素时,就可以使用内联样式。使用内联样式的方法是在相关的标签中使用样式属性。样式属性可以包含任何 CSS 属性。代码格式为:

```
<标记符  style = "property:value1;property:value2; … property:value">
```

【例 2.16】 使用内联样式实例。

代码如下。

```
< h1 style = "font - family:楷体_gb2312; text - align:center;font - size:56px">
   使用 h1 样式
</h1 >
< p style = "font - size:24px; text - align:center">
  在段落标签标记符中直接嵌套样式
</p >
< p >该段落标签没有使用样式</p >
```

运行效果如图 2.16 所示。

图 2.16 使用内联样式

2. 内部样式表

当单个文件需要特别样式时,就可以使用内部样式表。可以在 head 部分通过<style>标签定义内部样式表。代码格式为:

```
selector{property:value1;property:value2; … }
```

【例 2.17】 内部样式表实例。

代码如下。

```
< head >< title >在 STYLE 标记符中定义样式信息</title >
    < style >
      <! --
         P{ font - size:44px; text - align:center; }
         H1{ font - family:楷体_gb2312; text - align:center;color:red }
      -- >
    </style >
</head >
< body >
  < H1 >第一次使用 h1 样式</H1 >
  < P > STYLE 标记符中使用定义样式,该段落使用样式</P >
  < hr color = red >
  < H1 >第二次使用 h1 样式</H1 >
</body >
```

运行效果如图 2.17 所示。

收藏夹 在STYLE标记符中定义样式信息

第一次使用h1样式

STYLE标记符中使用定义样式,该段落使用样式

第二次使用h1样式

图 2.17 使用内部样式

3. 外部样式表

当样式需要被应用到很多页面的时候,外部样式表将是理想的选择。使用外部样式表,可以通过更改一个文件来改变整个站点的外观。这时可以将重复在多个网页中使用的样式放在外部样式表文件中,然后通过链接的方式引用其中的样式。步骤如下。

1) 创建外部样式表

将样式表定义放置到一个空白的文件中,然后将文件保存为后缀为.css 的文件。

2) 在网页中引用样式

引用样式的格式为:

```
< link   rel = "stylesheet"   type = "text/css"   href = "样式表文件">
```

【例 2.18】 外部样式表实例。

(1) 新建名称为"mycss.css"的文件,文件代码如下。

```
P{font - size:44px; text - align:center;color:blue }
H1{ font - family:楷体_gb2312; text - align:center; color:red}
```

(2) 新建名称为"样式练习(链接外部样式).html"的文件,文件中代码如下。

```
< html >
```

```
    <head><title>链接外部样式</title>
    <link rel = "stylesheet" type = "text/css" href = "mycss.css">
    </head>
    <body>
        <h1>第一次使用 h1 样式</h1>
        <p>链接外部样式,该段落标签中使用</p>
        <hr color = "red">
        <h1>第二次使用 h1 样式</h1>
    </body>
</html>
```

运行效果如图 2.18 所示。

收藏夹 链接外部样式

第一次使用h1样式

链接外部样式,该段落标签中使用

第二次使用h1样式

图 2.18 使用外部样式表

小结

本章首先介绍了 HTML 的基本概念,通过一个简单的 HTML 实例,读者可以了解 HTML 基本结构,如何编写 HTML。接着对 HTML 标签的概念、分类以及 HTML 元素概念做简要介绍,最后结合实例详细介绍了 HTML 标签及其使用,包括 HTML 文件基本标签、标题标签、字体标签、格式控制标签、图片标签、链接标签、表格标签、框架标签、列表标签、表单标签、多媒体标签、块标签、样式标签。本章所讲的 HTML 基本知识基本上涵盖了要求掌握 HTML 的全部重要内容。

通过本章的学习,读者能够掌握最基本的网页制作技术,为后面学习动态网页知识打下基础。

习题

1. 简述 HTML 文档的基本结构。
2. 简述 HTML 中每个标签的含义。
3. 利用 HTML 制作一个每隔 1min 自动刷新的网页。
4. 利用 HTML 制作包含不同超链接的网页。
5. 利用 HTML 制作一个带合并单元格的表格网页。
6. 利用 HTML 制作一个会员注册网页。
7. 利用 HTML 制作一个带嵌套的框架网页。
8. 利用 HTML 在网页中嵌入不同的多媒体元素,例如图像、动画、视频等。

第3章 Dreamweaver CS6基本操作

本章要点：

- Dreamweaver CS6 概述；
- 站点的创建及管理；
- 制作简单网页；
- 创建链接；
- 其他的一些网页元素；
- 表单。

Dreamweaver 是目前使用率较高的构建网站的专业工具，它提供的可视化布局、应用程序开发等诸多功能，使各级别的开发者都能快速创建网站。本章主要介绍 Dreamweaver CS6 的基本使用方法。

3.1 Dreamweaver CS6 概述

Dreamweaver 是美国 Macromedia 公司开发（后被 Adobe 公司收购）的集网页制作和管理网站于一身的所见即所得的网页编辑器。

Dreamweaver CS6 相较之前的版本，增强了许多新的功能。

1. 流体网格布局

使用基于 CSS3 的自适应网格版面系统，来创建跨平台和跨浏览器的兼容网页设计。利用简洁、业界标准的代码为各种不同设备和计算机开发项目，提高工作效率。它可直观地创建复杂网页设计和页面版面，无须忙于编写代码。

2. 改善的 FTP 性能

利用重新改良的多线程 FTP 传输工具节省上传大型文件的时间，更快速高效地上传网站文件，缩短制作时间。

3. 增强型 jQuery Mobile 支持

使用更新的 jQuery Mobile 支持为 iOS 和 Android 平台建立本地应用程序。建立触及移动受众的应用程序，同时简化开发者的移动开发工作流程。

4. 更新的 PhoneGap 支持

更新的 Adobe PhoneGap 支持可轻松为 Android 和 iOS 建立和封装本地应用程序。通过改编现有的 HTML 代码来创建移动应用程序。使用 PhoneGap 模拟器检查开发者的设计。

5. CSS3 转换

将 CSS 属性变化制成动画转换效果,使网页设计栩栩如生。在处理网页元素和创建优美效果的同时保持对网页设计的精准控制。

6. 更新的实时视图

使用更新的"实时视图"功能在发布前测试页面。"实时视图"现已使用最新版的 WebKit 转换引擎,能够提供绝佳的 HTML5 支持。

7. 更新的多屏幕预览面板

利用更新的"多屏幕预览"面板检查智能手机、平板电脑和台式计算机所建立项目的显示画面。该增强型面板现在能够让开发者检查 HTML5 内容呈现。

3.1.1 Dreamweaver CS6 的工作区

选择"开始"|所有程序|Adobe Dreamweaver CS6 命令,即可启动 Dreamweaver CS6 软件。

首次启动 Dreamweaver CS6 时,会弹出"默认编辑器"对话框,用户可根据自己的需要和喜好设置默认的编辑器。然后单击"确定"按钮,将打开 Dreamweaver CS6 工作区的开始页面,如图 3.1 所示。默认情况下,Dreamweaver CS6 的工作区布局是以"设计"视图布局的。

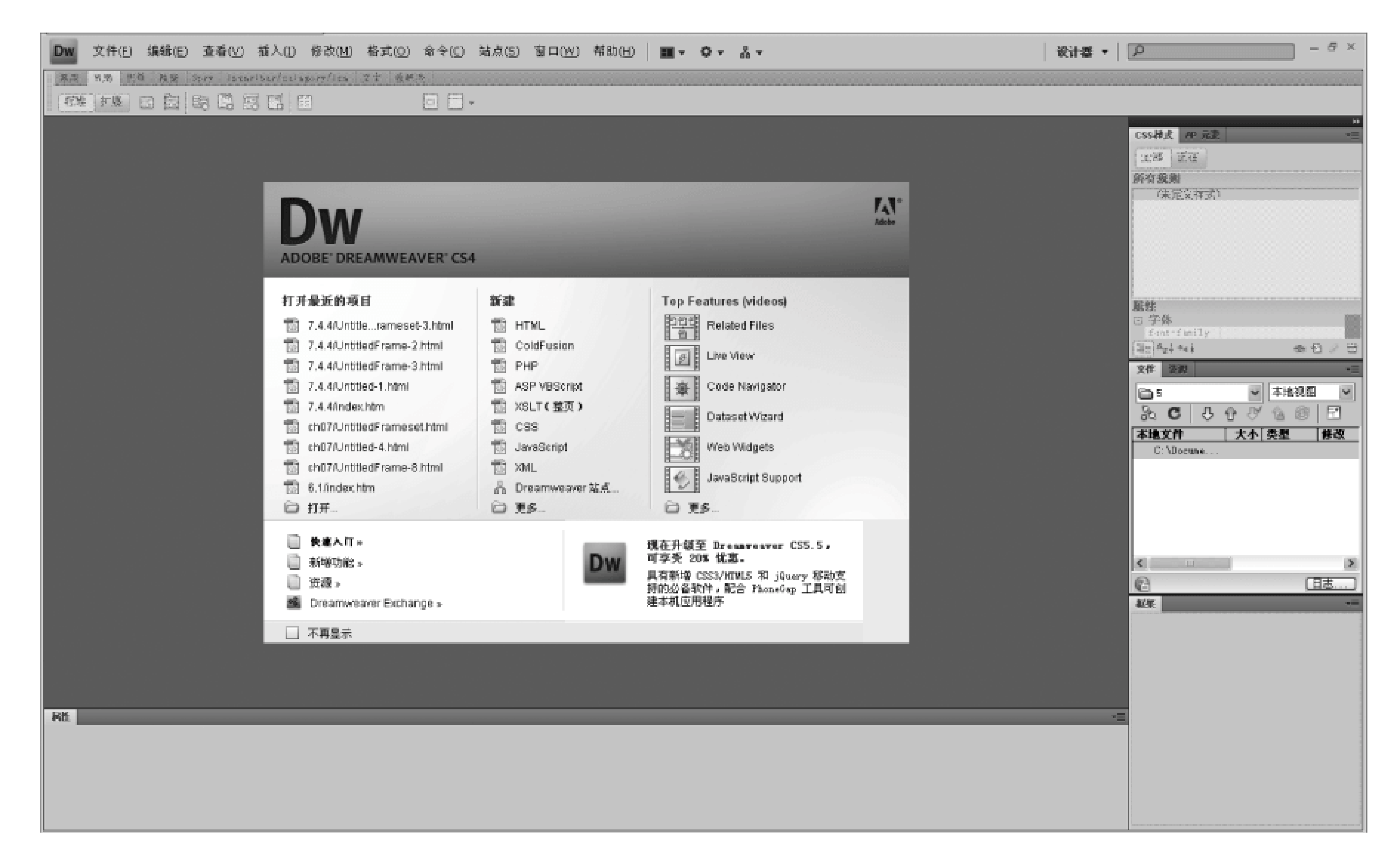


图 3.1　Dreamweaver CS6 工作区的开始页面

Dreamweaver CS6 工作区的开始页面由以下三列组成。

(1) 打开最近的项目：此栏中列出了最近打开过的文件列表，可以通过双击文件快速将其打开。

(2) 新建：此栏中列出了可以创建的新文件类型。如果没有找到需要创建的文件类型，可以单击下方的"更多"选项，调出更多可创建的文件类型。

(3) 主要功能：展示了 Dreamweaver CS6 的主要功能。

在开始页面中，单击"新建"|HTML 选项，将打开 Dreamweaver CS6 的工作界面，如图 3.2 所示。



图 3.2 Dreamweaver CS6 的工作界面

Dreamweaver CS6 的工作界面主要包括"应用程序栏"、"菜单栏"、"插入栏(插入面板)"、"文档工具栏"、"标签选择器"、"状态栏"、"属性检查器"、"面板组(标签组)"、"文档窗口"。

(1) 应用程序栏：主要包含一个工作区切换器，方便不同用户根据个人的使用习惯选择不同的页面布局；Dreamweaver CS6 扩展按钮；一个站点管理按钮。

(2) 菜单栏：包含 10 项主菜单，几乎涵盖了 Dreamweaver CS6 中的所有功能。通过菜单栏的使用可以对对象进行任意的操作与控制。菜单栏按照功能的不同进行了相应的划分，使用户在使用的时候能够更方便。

(3) 插入栏：主要用于在网页中插入各种网页元素，如文字、图像、表格、按钮、导航以及程序等。

(4) 文档工具栏：通过单击文档工具栏中的各种视图按钮，用户可以自由地在不同的视图之间快速切换。另外，文档工具栏还包含一些与查看文档、在本地和远程站点间传输文档有关的常用命令和选项。

(5) 标签选择器：位于文档窗口底部的状态栏中，用于显示当前选定对象 HTML 标签的层次结构。单击该层次结构中的任何标签，就可以选择该标签及其全部内容，不过标签选

择器在代码视图中是不可见的。

(6) 状态栏：其作用是显示当前正在编辑文档的相关信息，如当前窗口大小、文档大小和估计下载时间等。

(7) 属性检查器：通常用于设置和查看所选对象的各种属性，位于窗口底部。单击“属性”标题栏可展开或折叠“属性”面板。该面板中显示的属性通常对应标签的属性，更改属性通常与在代码视图中更改相应的属性具有相同的效果。

(8) 面板组：为使设计界面更加简洁，同时也为了获得更大的操作空间，Dreamweaver CS6 中将类型相同或功能相近的面板分别组织到不同的面板下，再将这些面板组织在一起，构成面板组。面板组中的面板都是折叠的，单击标题左角处的展开箭头可以折叠或展开面板，并且可以和其他面板组停靠在一起。面板组还可以停靠到集成的应用程序窗口中。

(9) 文档窗口：即设计区，是 Dreamweaver CS6 进行可视化编辑网页的主要区域，可以显示当前文档的所有操作效果，如插入文本、图像、动画等。

3.1.2 Dreamweaver CS6 的插入栏

插入栏位于菜单栏和文档工具栏的中间，它涵盖了在设计网页时最常用的项目，就好像在绘画调色时使用的三原色一样。在插入栏中包含用于创建和插入对象(如表格、AP Div 和图像)的按钮，而这些按钮则又被组织到各个选项卡中，如图 3.3 所示。



图 3.3 插入栏

插入栏被分为独立的对象类别：常用、布局、表单、数据、Spry、InContext Editing、文本、收藏夹。

1. “常用”选项卡

如图 3.3 所示，其中为用户准备了最常用的插入对象，如表格、图像。

2. “布局”选项卡

如图 3.4 所示，用于处理表格、DIV 标签、AP Div 和框架，通过这些对象可以定义页面布局。Dreamweaver CS6 提供了两种方式来使用表格，分别是标准视图和扩展视图。另外，还可以进行单元格的布局和表格的布局。



图 3.4 “布局”选项卡

3. “表单”选项卡

如图 3.5 所示，它是实现 HTML 互动性的一个主要方式。“表单”选项卡中为用户提

供了用来创建基于网页表单的基本构建块。



图 3.5 “表单”选项卡

4. “数据”选项卡

如图 3.6 所示，主要用来添加与网站后台数据库相关的动态交互元素，例如记录集、重复区域以及插入表单和更新记录表单等。



图 3.6 “数据”选项卡

5. Spry 选项卡

如图 3.7 所示，Spry 选项卡是一个 JavaScript 库，网页设计人员使用它可以创建能为站点访问者提供更多丰富体验的网页。在 Dreamweaver CS6 中，又新增了包括 XML 驱动的列表和表格、折叠构件、选项卡式面板、Spry 工具提示等元素。



图 3.7 Spry 选项卡

6. InContext Editing 选项卡

如图 3.8 所示，InContext Editing 是 Dreamweaver CS6 新增的用户界面对象，包含供生成 InContext 编辑页面的按钮，分别是创建重复区域、创建可编辑区域、管理可用的 CSS 类。通过该功能，网页设计人员不但可以维护其站点设计的完整性，同时还允许客户处理他们自己的更新。



图 3.8 InContext Editing 选项卡

7. “文本”选项卡

如图 3.9 所示，该选项卡包含最常用的文本格式 HTML 标签，如强调文本、改变文本字体或创建项目列表所需要的选项。文本类别包含一个字符按钮和一些特殊字符，还包含可以插入换行符和不间断空格的对象。



图 3.9 “文本”选项卡

8. “收藏夹”选项卡

用户可以从插入栏中选择自己常用的工具放入其中,以提高工作效率。

3.2 站点的创建及管理

3.2.1 站点概述

站点是一种管理网站中所有相关联文档的工具,通过站点可以实现将文件上传到网络服务器,自动跟踪和维护、管理文件以及共享文件等功能。严格来说,站点也是一种文档的组织形式,由文档和文档所在的文件夹组成,不同的文件夹保存不同的网页内容,如 images 文件夹用于存放图片,这样便于以后的管理与更新。

Dreamweaver 中的站点包括本地站点、远程站点与测试站点。本地站点是用来存放整个网站框架的本地文件夹,是用户的工作目录,一般制作网页时只需要建立本地站点即可。远程站点是存储于 Internet 服务器上的站点和相关文档。通常情况下,为了不连接 Internet 而对所建的站点进行测试,可以在本地计算机上创建远程站点,来模拟真实的 Web 服务器进行测试。测试站点是 Dreamweaver 处理动态页面的文件夹,使用此文件夹生成动态内容并在工作时连接到数据库,用于对动态页面进行测试。

3.2.2 创建本地站点

创建站点是制作网站的第一步,可以更好地利用站点窗口对站点文件进行管理,也可以尽可能地减少错误的出现。在定义站点的过程中,文件夹的命名尤为重要,不能用中文,否则会在上传网站时带来许多不必要的麻烦。

创建本地站点时,首先要在本地计算机的磁盘中建立相应的物理目录(即指定文件夹)。例如,在本地计算机的本地磁盘(E:)中创建好一个 Mysit 文件夹,在其中再创建相应的 images、music、flash 等子文件夹。建立好目录后,即可用该目录作为 Dreamweaver 站点的本地根文件夹。

1. 使用站点向导创建站点

使用站点定义向导可以带领用户逐步完成站点的设置操作,这种创建方法适合于初学者。

在菜单栏执行“站点”|“新建站点”命令,在打开的“定义站点”对话框中,用户可以在“基本”选项卡和“高级”选项卡中任意选择一种来填写。“基本”选项卡可以指导用户一步一步地完成站点的设置。“高级”选项卡则比较适用于对 Dreamweaver 比较熟悉的用户,使用

“高级”选项卡就没有了站点定义向导，完全按照用户自己的意愿进行。

在这里先使用“基本”选项卡，如图3.10所示。

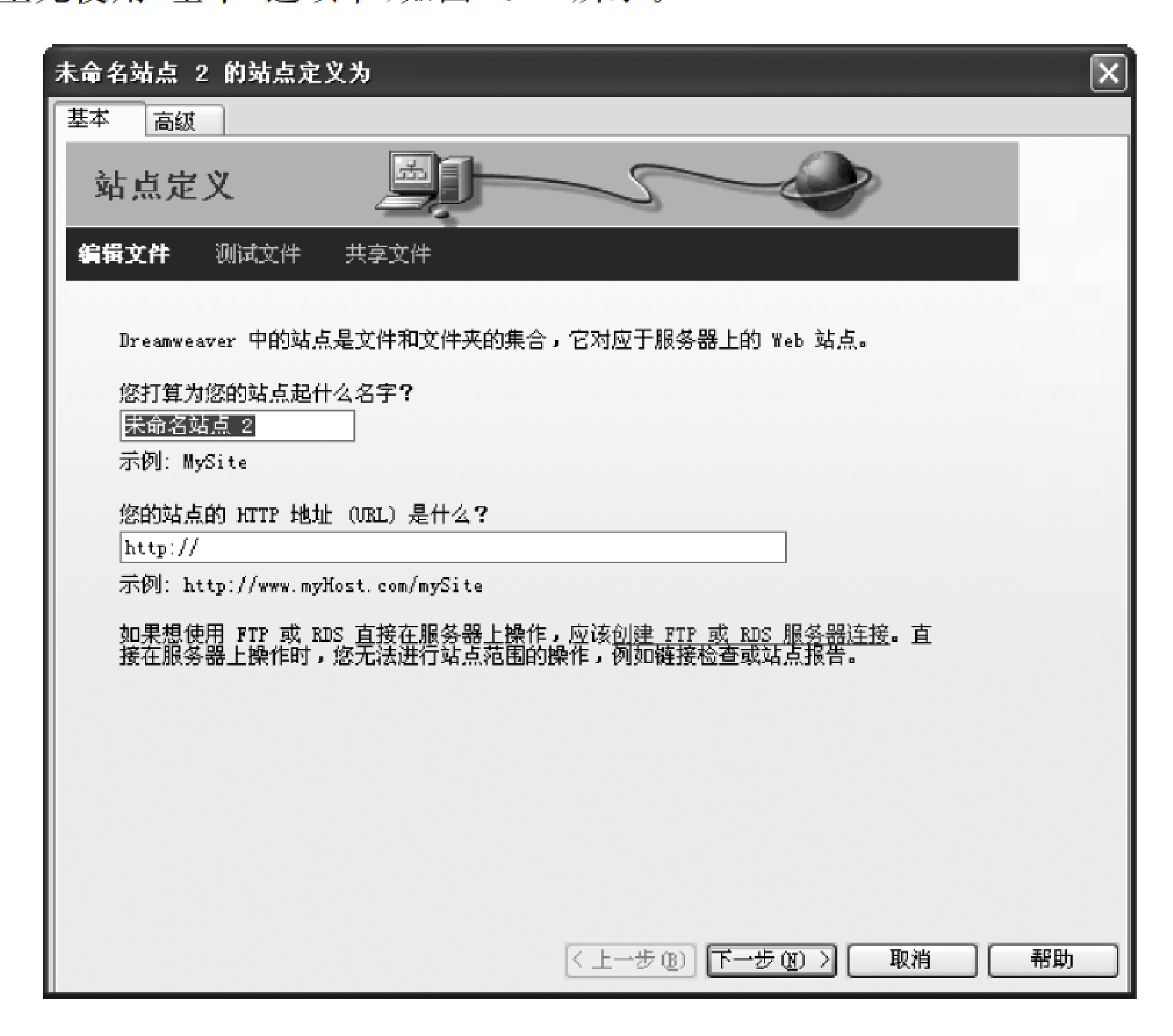


图3.10 新建站点

在“您打算为您的站点起什么名字？”的问题下面输入这个站点的名称。默认情况下名字为“未命名站点1”、“未命名站点2”……，可以根据需要来设置自己的站点名称，也可以暂时不设置，在以后“管理站点”时再次修改站点名称。这里输入站点名称“mysit”。

然后单击“下一步”按钮，弹出如图3.11所示对话框。在该对话框中会询问是否使用服务器技术。如果选择“是，我想使用服务器技术”，那么会在下面多出来一个服务器种类选择的项目，在这里可以选择合适的服务器。如果当前用户建立的站点暂时不使用Web应用程序，建议在这一步中不要选择使用服务器技术。现在要建立的是静态站点，故选择“否，我不想使用服务器技术”选项。

继续单击“下一步”按钮，又会弹出一个对话框，如图3.12所示。在一般情况下，制作好的网站都要上传到网络上，供浏览者浏览。但上传到网络上还需要虚拟主页的空间，所以一般是先在本地制作网站，最后上传给服务器。这里选择“编辑我的计算机上的本地副本，完成后再上传到服务器（推荐）”，接着再为mysit站点找一个本地根文件夹。

注意：创建本地站点时，首先应在本地计算机的磁盘中建立相应的物理目录（即指定文件夹）。

例如，在本地计算机的本地磁盘（E:）中创建mysit文件夹，在此文件夹中再创建子文件夹images用来存放网站制作中相关的图像。建立好目录后，即可用该目录作为Dreamweaver站点的本地根文件夹。

继续单击“下一步”按钮。在这个对话框中可以在“您如何连接到远程服务器？”下拉选



图 3.11　是否使用服务器



图 3.12　是否上传服务器

项中设置连接远程服务器的方法，如图 3.13 所示。由于上一步已经设定了暂时不连接服务器，故这里也选择“本地/网络”，即现在仅在本地计算机上进行制作。完成后可以再次设置服务器连接方式。



图 3.13 如何连接远程服务器

完成后单击“下一步”按钮，接着会弹出“总结”对话框，如图 3.14 所示。显示当前创建的这个站点的本地信息，测试服务器的类型等。在这个对话框中不需要进行任何选择，单击“完成”按钮，这个站点的初始化工作就完成了。

2. 使用“高级”选项卡创建站点

使用站点向导定义站点主要是针对初学者，而对熟悉 Dreamweaver 的人来说，一般都采用“高级”选项卡创建站点。

首先在菜单栏上执行“站点”|“管理站点”命令，或者单击“文件”面板中的“站点”下拉列表，从中选择“管理站点”项。同样，在 Dreamweaver CS6 界面中，可以直接单击程序栏上新增的 品 图标，在弹出菜单中选择“管理站点”选项。在“管理站点”对话框中单击“新建”按钮，选择“站点”选项后会弹出一个如图 3.15 所示的对话框，在该对话框中选择“高级”选项卡。

在这个选项卡中输入用户的站点名称“mysit”，接着设置本地根文件夹和默认图像文件夹，其他的选项和设置使用默认即可。

除了确定本地站点根目录，如果还需要使用和远程站点有关的信息，可以在“分类”栏中选择“远程信息”项，如图 3.16 所示。在它的选项中包括 FTP、本地/网络、WebDAV 和 RDS 等设置。

然后在“高级”选项卡左边的“分类”栏中选择“测试服务器”选项，如图 3.17 所示。对“服



图 3.14 "总结"对话框



图 3.15 建立站点



图 3.16 “远程信息”选项卡



图 3.17 测试服务器选项卡

务器模型”选项的设置也和之前选择服务器一样，在这里暂时选择“无”。在“访问”选项中选择“本地/网络”。在“URL 前缀”文本框中，输入测试服务器上根文件夹的 HTTP 地址。如果在本地计算机上工作，URL 前缀是以 http://localhost/开始。最后，单击“确定”按钮，站点创建完毕。

3.2.3 管理站点

设置好站点后，可以对本地站点进行多方面的管理，如打开站点、编辑站点、删除站点及复制站点等。

在菜单栏中选择“站点”|“管理站点”命令，打开“管理站点”对话框，如图 3.18 所示。



图 3.18 “管理站点”对话框

在该对话框中选择需要修改的站点，然后单击“编辑”按钮，之后会再次打开相关站点的定义对话框，此时可以对站点信息进行编辑，编辑站点的过程与新建站点相同，如图 3.19 所示。

在制作网站过程中，如果仅需要更改某个站点里的部分页面内容，而不破坏原站点内容时，可以先将该站点复制出来，然后在该站点的副本上修改。单击“复制”按钮，在“管理站点”对话框中会直接出现该站点的副本，如图 3.20 所示。



图 3.19 编辑站点



图 3.20 “管理站点”对话框

一个项目中的任务结束后，可以从列表上删除站点。在“管理站点”对话框中，选择要删除的站点，然后单击“删除”按钮即可。需要注意的是，删除站点只是把站点从 Dreamweaver 的内部站点列表上删除，并没有从硬盘上真正删除任何的文件或文件夹。以后如果有需要，还可以再次将其设置为站点。

删除站点之前，需要把站点设置导出，用户可以在“管理站点”对话框中单击“导出”按钮，会出现“导出站点”对话框，如图 3.21 所示。选择合适的保存目录，然后单击“保存”按钮，即可完成站点的导出。导出后的文件保存了所有的链接信息，并能够在以后通过“管理站点”对话框中的“导入”按钮再次将其导入。



图 3.21 “导出站点”对话框

不管是新建、复制、删除、导入还是导出站点，完成操作后单击“完成”按钮会自动关闭“管理站点”对话框。

3.3 制作简单网页

浏览网页时，通过文本和图像获取信息是最直接的方式。文本是基本的信息载体，不管网页内容如何丰富，文本始终都是网页中最基本的元素。而图像使得网页的内容更加丰富多彩，形象生动。掌握好文本和图像的使用方法是设计制作网页的基本能力。

3.3.1 文档的基本操作

1. 创建空白文档

制作网页应该从创建空白文档开始。创建空白文档的具体步骤如下。

(1) 选择"文件"|"新建"菜单命令，打开"新建文档"对话框，如图 3.22 所示。



图 3.22 "新建文档"对话框

(2) 选择"空白页"选项，在"页面类型"列表中选择 HTML 选项，在右侧列表框中选择"无"选项。

(3) 单击"创建"按钮，即可创建一个空白文档。

2. 页面属性设置

创建空白文档后，要对文件进行页面属性的设置，也就是设置整个网站页面的外观效果。选择"修改"|"页面属性"菜单命令，或按 Ctrl+J 组合键，打开"页面属性"对话框，从中

可以设置外观、链接、标题、标题/编码和跟踪图像等属性，如图 3.23 所示。

图 3.23 设置页面属性

3.3.2 文本及文本属性设置

网页中，文本是基本的信息载体，是网页中最基本的元素之一。

1. 插入文本

（1）将光标放置在文件窗口中，输入文字。

（2）选择“文件”|“另存为”菜单命令，弹出“另存为”对话框。在“保存在”下拉列表中选择文件要保存的路径，在“文件名”文本框中输入文件名，在“保存类型”下拉列表中选择“HTML 文档”选项，单击“保存”按钮保存文件。

（3）按 F12 键在浏览器中预览效果。

注意：在输入文本的过程中，换行时如果直接按 Enter 键，行间距会比较大。一般情况下，在网页中换行时按 Shift+Enter 组合键，这样才是正常的行距。

2. 文本属性的设置

1）设置文本标题

首先在文件窗口中选中要设置标题的文字，然后在菜单栏中执行“修改”|“页面属性”命令，打开“页面属性”对话框。

选择“分类”栏里的“标题(CSS)”选项，在标题的设置项中可以看到，文本标题的设置共有 6 个级别，如图 3.24 所示。

在对这些标题进行设置时，每个标题的字体大小和颜色都可以单独设置。需要对所有标题设置字体类型。通常情况下“标题字体”列表中会为用户列出一些默认的字体，如果这些字体中没有需要的字体，可以在“标题字体”下拉列表中选择最下面的“编辑字体列表”，进而在“编辑字体列表”对话框中添加或删除字体类型，如图 3.25 所示。

首先在“可用字体”中选择字体，接着单击“添加”按钮。这样就可以将选中的字体添加



图 3.24 设置标题属性



图 3.25 “编辑字体列表”对话框

到标题字体中。当然,在字体列表中还可以对当前已有的字体进行添加或删除,只需单击字体列表左上角的“加号”和“减号”按钮就可以管理字体。要注意的是,每次从可选字体中添加字体只能添加一个;如果超过了一个,那么超出的字体将和第一个添加的字体列在同一项中。

注意: 用户在添加字体类型的时候,最好使用宋体、楷体、仿宋和黑体这 4 种常用字体。如果使用一些不常用的字体,当网页在用户端显示时,如果用户端的字体库中没有当前字体,那么将无法正常显示。

2) 添加空格

默认状态下,Dreamweaver CS6 中不能连续输入多个空格,只能输入一个,继续按空格键光标将不会再向后移动。如果想要连续输入多个空格,方法很多,最简单的就是将输入法提示框上的“半角”改为“全角”。或者按 Shift+Ctrl+Space 组合键增加连续空格。

3) 在属性检查器中编辑 CSS 规则

在属性检查器中,开发人员用两个标签将 HTML 格式和 CSS 格式完全分开,设计者可以有选择地设置文本属性。当单击 HTML 格式时,属性面板中呈现与 HTML 相关的选

项，设置完成后，Dreamweaver 会将属性添加到页面正文的 HTML 代码中。同样，如果应用了 CSS 格式，Dreamweaver 就会将属性写入文档头或单独的样式表中。

首先在文档窗口中选定要进行设置的文本。然后打开“属性”窗口（通过执行菜单命令“窗口”|“属性”或者按 Ctrl+F3 快捷键），单击 CSS 按钮，如图 3.26 所示。



图 3.26 CSS 属性

“目标规则”：是指在 CSS 属性检查器中正在编辑的规则。当文本已应用了样式规则时，在页面的文本内部单击，将会显示出影响该文本格式的规则。如果要创建新规则，在“目标规则”下拉列表中选择“新 CSS 规则”，完成新建 CSS 规则。

“编辑规则”按钮：用以打开目标规则的“CSS 规则定义”对话框。

“字体”：更改目标规则的字体。

“大小”：设置目标规则的字体大小。

文本颜色：可以将所选颜色设为目标规则的字体颜色。可以通过单击颜色框选择 Web 安全色，或在相邻的文本字段中输入颜色值。

粗体：用于给目标规则添加粗体属性。

斜体：用于给目标规则添加斜体属性。

“左对齐、居中对齐和右对齐”：用来设置目标规则的各种对齐属性。

4）在属性检查器中设置 HTML 格式

选择要设置格式的文本，打开属性检查器，单击 HTML 按钮。设置要应用于所选文本的选项，如图 3.27 所示。



图 3.27 HTML 属性

“格式”：用于设置所选文本的段落样式。

ID：用于为所选内容分配 ID，以表示其唯一性。ID 默认情况下为无，如果适用，下拉列表中将列出文档内所有未适用的已声明 ID。

“类”：用于显示当前应用于所选文本的类样式。如果没有对所选内容应用过任何样式，则显示无。如果对所选内容应用了样式，则下拉列表中会显示出应用于该文本的样式。

项目列表：用于为所选文本创建项目列表。又被称为无序列表，有方形、空心圆形和实心圆形三种表示标记方式。

编号列表：用于为所选文本创建编号列表。又被称为有序列表，可以用数字、大小写字母、大小写罗马数字来标记。

文本缩进和凸出：用于通过应用或删除 blockquote 标签，减小所选文本或删除所选文

本的缩进。在列表中，运用缩进可以创建一个嵌套列表，而与它有相反作用的“凸出”则会删除缩进以取消嵌套列表。

“链接”：用于创建所选文本的超文本链接。

“标题”：用于为超级链接指定文本工具提示。

“目标”：用于指定将链接文档加载到哪个框架或窗口。

3.3.3 插入图像及图像属性设置

网页中常用的图像格式有三种，分别是 GIF、JPEG 和 PNG。下面简单介绍它们的特点。

(1) GIF 图像是一种图形交换格式，是一种位图图像，最多可以显示 256 种颜色。它占用磁盘空间小，支持透明背景和动画，多数用于图标、按钮、卡通图像、滚动条和背景等。

(2) JPEG 是一种图像压缩格式，支持约 1670 万种颜色。它没有透明度和动画的功能。主要用于摄影图片，一般用来表示真实的事物。

(3) PNG 格式是近几年开始流行的一种格式。它在压缩方面能够像 GIF 一样没有像素上的损失，还能像 JPEG 那样呈现更多的颜色，在显示速度上比 GIF 和 JPEG 更快一些，同时还支持透明度和动画。

在网页中插入图像的具体步骤如下。

(1) 在设计视图下，将光标放置在要插入图像的位置。在“插入”面板的“常用”选项卡中单击图标。

(2) 弹出“选择图像源文件”对话框，从中选择要插入的图像文件。

(3) 单击“确定”按钮即可完成图像的插入。

插入图像后，选定图像，可以在“属性”面板中设置图像的属性。“属性”面板如图 3.28 所示。

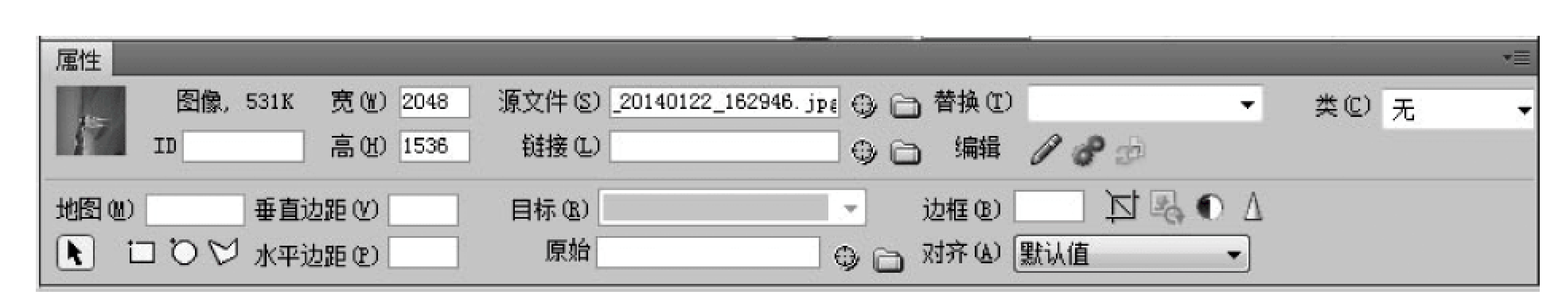


图 3.28 图像“属性”面板

“地图”：用于创建图像的热区。

热点工具按钮：单击这些按钮，可以创建图像的热点链接。

“宽”和“高”：设置在浏览器中显示图像的宽度和高度，以像素为单位。如果要调整图像的大小，可以在“宽”和“高”文本框中输入相应的数值，调整后，其文本框的右侧将显示“重设图像大小”按钮，单击该按钮，可恢复图像到原来大小。

“垂直边距”和“水平边距”：分别设置图像顶部、底部的边距和左右的边距。

“源文件”：指定图像的路径，单击文本框右侧的“浏览”按钮，可弹出“选择图像源文件”对话框，从中可选择图像文件。

“链接”：用于指定图像的链接文件。可拖动“指向文件”图标到“文件”面板中的某个

文件上实现链接。

“目标”：用于指定链接页面在框架或窗口中的打开方式。其下拉列表中有以下 4 个选项。

（1）_blank：在弹出的新浏览器窗口中打开链接文件。

（2）_parent：如果是嵌套的框架，会在父框架或窗口中打开链接文件；如果不是嵌套的框架，则与_top 相同，在整个浏览器窗口中打开文件。

（3）_self：在当前网页所在的窗口中打开链接，此目标为浏览器默认设置。

（4）_op：在完整的浏览器窗口中打开链接文件。

“原始”：设置图像下载完成前显示的低质量图像。

“替换”：设置图像的说明性文字，用于在浏览器无法显示图像时替代图像显示的文本。

“编辑”：启动图像编辑器中的一组编辑工具对图像进行复杂的编辑。

“边框”：设置图像边框的宽度，单位为像素。

“对齐”：设置图像和文本的对齐方式。

3.4 创建链接

链接是网页中极为重要的部分，单击文档中的链接，即可跳转至相应的位置，正是因为有了链接，才可以在网站中相互跳转而方便地浏览各种各样的信息，享受网络带来的无穷乐趣。在正式学习链接前，首先需要了解链接中的路径。在计算机中，每个文件都有存放位置和路径，这些文件的路径和存放位置和用户制作链接是紧密相连的。

链接中共有三种路径：绝对路径、相对路径、根路径。通常情况下，在添加外部链接时，使用绝对路径；为网页添加内部链接时，用根路径和文件的相对路径。

补充：链接分为内部链接和外部链接。它们都是相对于站点文件夹而言的。如果添加的链接是指向站点文件夹之内的文件，即为内部链接；如果添加的链接是指向站点文件夹之外的文件，即为外部链接。

1. 绝对路径

绝对路径是为文件提供的完整路径，其中包括适用的协议，常见的有 HTTP、FTP 等。例如，http://www.sina.com，在这个路径中不仅有网页的地址，还附带有它所适用的协议。当用户制作的链接要连接到其他网站中的文件时，必须使用绝对路径。

2. 相对路径

相对路径适用于内部链接。凡是属于同一网站之下的文件，就算不在同一个目录下，也可以使用相对路径。即：只要是处于站点文件夹之内，相对地址可以自由地在各个文件之间构建链接。因为这种地址使用的是构建链接的两个文件之间的关系，不会受到站点所处的服务器位置的影响，故可以省略绝对地址中的相同部分。这种方式可以保证在站点文件夹所在的服务器地址发生改变的情况下，文件夹的所有内部链接都不会出现错误或无法链接。

3. 根路径

根路径也适用于创建内部链接，只有站点的规模非常大，需要放置在几个服务器上或是在一个服务器上放置多个站点时才使用。

根目录相对地址在书写时要以“/”开头，代表根目录，然后再在它的后面添加上文件夹名和文件名，按照它们的从属顺序书写。根路径以“/”开头，后面则是根目录下的目录名(例如：/html/index-1.html)。

3.4.1 文本和图像链接

浏览网页时，会看到一些带下划线的文字，将鼠标移动到文字上时，鼠标指针将变成手形，单击鼠标会打开一个网页，这样的链接就是文本链接。若将鼠标移到图像上，鼠标指针变成手形，单击鼠标会打开一个网页，这样的链接就是图像链接。

通过 Dreamweaver 可以方便地创建文本链接和图像链接，可以使用站点地图查看、编辑和删除链接。

创建文本链接的常用方法有如下两种。

(1) 在浏览视图中，选定需要创建文本链接的文本。单击“属性”面板中的“浏览文件”图标，位置如图 3.29 中 1 所示。打开“选择文件”对话框，在该对话框中选择要链接的网页文件，单击“确定”按钮即可。

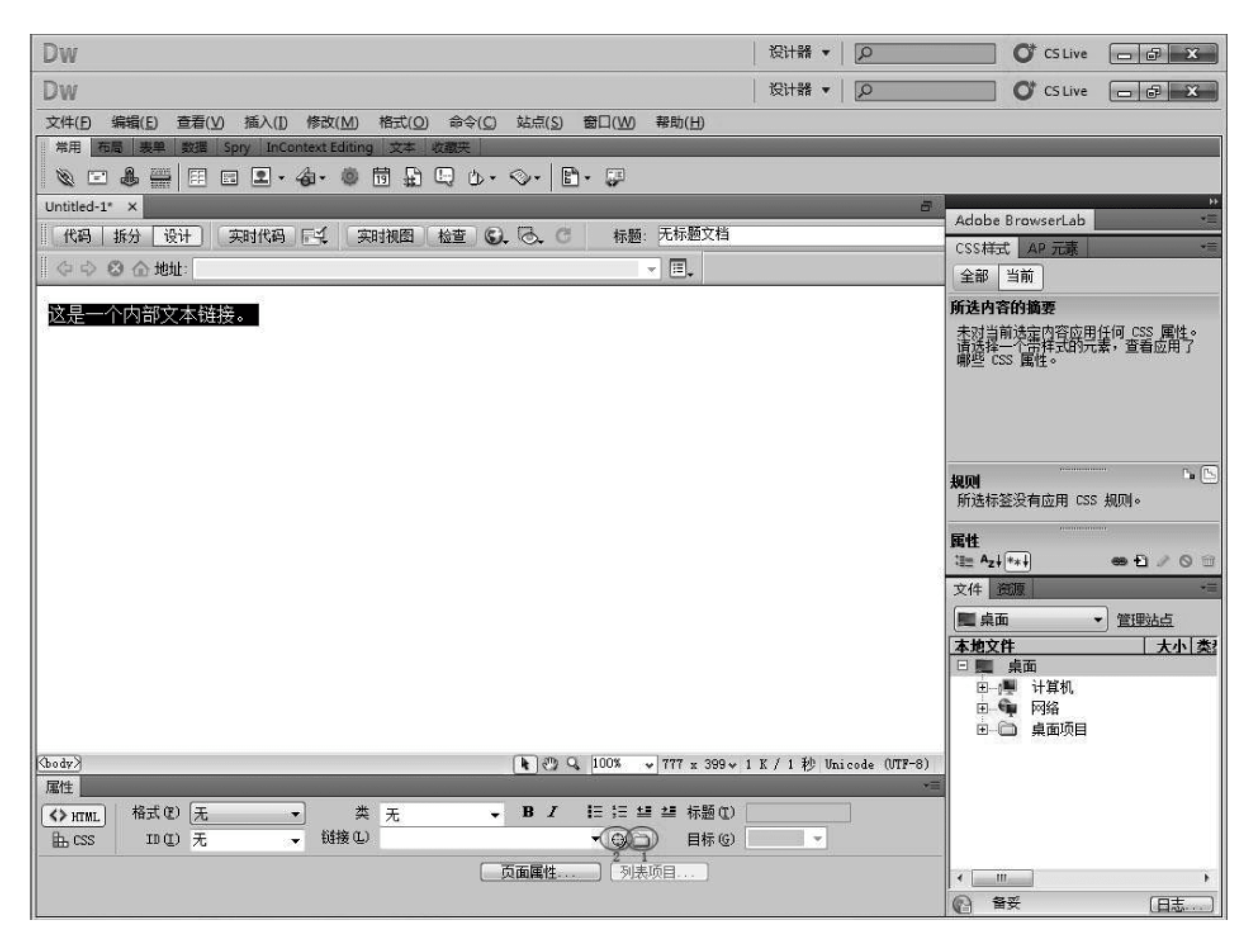


图 3.29 浏览视图中创建链接

(2) 在浏览视图中,选定需要创建文本链接的文本。单击"属性"面板中的"指向文件"图标,位置如图3.29中2所示,将其拖动到右面"文件"面板中要链接的网页上,然后松开鼠标,即可完成文本链接的建立。

创建图像链接的方法与创建文本链接的方法类似,常用的有如下两种。

(1) 在浏览视图中,选定需要创建图像链接的图像。单击"属性"面板中的"浏览文件"图标,位置如图3.29中1所示。打开"选择文件"对话框,在该对话框中选择要链接的网页文件,单击"确定"按钮即可。

(2) 在浏览视图中,选定需要创建图像链接的图像。单击"属性"面板中的"指向文件"图标,位置如图3.29中2所示,将其拖动到右面"文件"面板中要链接的网页上,然后松开鼠标,即可完成图像链接的建立。

以上创建的均是网站的内部链接,网站的外部链接创建方法与内部链接类似,需要注意的是外部链接的创建,必须使用绝对路径。例如,要在主页上添加新浪网站的图标,将它们与相应的网站链接起来。首先需要选中该图标,然后在"属性"面板中的"链接"文本框中输入新浪的网址 www.sina.com.cn 即可。

3.4.2 锚点链接

锚点链接可以使链接指向当前文档或不同文档中的指定位置,以便使访问者能够快速地浏览到选定的位置,加快信息检索的速度。

创建锚点链接首先要设置一个命名锚点,然后建立到命名锚点的链接。

创建命名锚点的步骤如下。

(1) 将光标放置在要命名锚点的位置,或选中要为其命名锚点的文本。

(2) 在"插入"面板的"常用"选项卡中单击"命名锚点"图标。

(3) 在弹出的"命名锚记"对话框中输入锚记名称(如 top),单击"确定"按钮,即可在文档窗口中相应位置看到锚记。

注意:在一篇文档中,锚记名称是唯一的,不允许在同一篇文档中出现相同的锚记名称。锚记名称中不能含有空格,不应置于层内。锚记名称区分大小写。

在文档中定义了锚记后,就可创建锚点链接,步骤如下。

(1) 选中作为锚点链接的文字(如网页中常见的"返回顶部")。

(2) 在"属性"面板的"链接"文本框中输入字符"#"和锚记名称,如"#top"。

(3) 保存文档,按F12键即可预览。

同样,命名锚记后,也可用"属性"面板中的"指向文件"图标来创建锚记链接,方法与文本链接类似,这里不再赘述。

3.4.3 电子邮件链接

电子邮件链接是一种特殊的链接,单击这种链接,不是跳转到相应的网页上,而是会启动计算机中相应的 E-mail 程序,允许书写电子邮件,然后发往指定地址。

创建电子邮件链接的步骤如下。

(1) 将光标置于文档窗口中要显示电子邮件链接的位置,或选中将显示为电子邮件链

接的文本或图像(如网页中常见的“联系我们”)。

(2) 在“插入”面板的“常用”标签下单击“电子邮件链接”图标。

(3) 在弹出的“电子邮件链接”对话框的“文本”文本框中,输入或编辑作为电子邮件链接显示在文档中的文本,在 E-mail 文本框中输入邮件送达的 E-mail 地址,然后单击“确定”按钮。

同样,也可以利用“属性”面板创建电子邮件链接。选定即将显示为电子邮件链接的文本或图像,在“属性”面板的“链接”文本框中输入 mailto:和电子邮件地址(如:mailto:wysjyzz@sina.com)。

(4) 保存文档,按 F12 键预览效果。

3.4.4 下载文件链接

下载文件链接的创建方法与一般链接的创建方法相同,只是所链接的内容不是文字或网页,而是一个软件。其步骤如下。

(1) 在文档窗口中,选中要设置为下载文件的链接的文本。

(2) 单击“属性”面板中的“浏览文件”图标,在打开的“选择文件”对话框中选择要链接的下载文件,然后单击“确定”按钮即可完成下载文件的链接的创建。

3.4.5 链接的检查

当创建好一个站点后,由于一个网站中的链接数量极多,因此在上传服务器前应先检查站点中所有链接是否有效。在 Dreamweaver 中,可以快速检查站点中网页的链接,其步骤如下。

(1) 菜单栏中选择“站点”|“检查站点范围的链接”菜单命令,此时会激活链接检查器。

(2) 从“属性”面板左上角的“显示”下拉列表中可以选择“断掉的链接”、“外部的链接”或“孤立文件”等选项。选取相应的选项,Dreamweaver 将对当前链接情况进行检查,并将相应的文件列表显示出来,如图 3.30 所示。

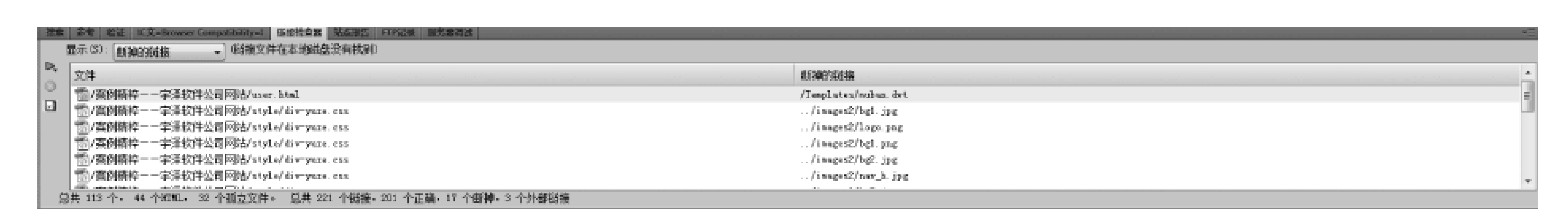


图 3.30 链接的检查

(3) 对于有问题的文件,直接双击即可将其打开进行修改。

3.5 其他的一些网页元素

网页中除了文本、图像、表格等这些基本元素外,还有一些元素也是非常重要的,如水平线、日期、特殊字符和多媒体元素等。

3.5.1 插入水平线

水平线主要用于分割文档内容，使文档结构清晰明了，便于浏览。在文档中插入水平线的方法如下。

（1）在文档中，将光标置于要插入水平线的位置。

（2）在菜单栏中选择“插入”|HTML|“水平线”命令，即可在文档窗口中插入一条水平线。

（3）选定插入的水平线，选择“窗口”|“属性”菜单命令，打开“属性”面板，如图3.31所示，在其中可进行以下属性的设置。



图3.31 水平线“属性”面板

“水平线”文本框：指定该水平线的ID，用于标识该对象。

“宽”和“高”：以像素为单位或以页面尺寸百分比的形式指定水平线的宽度和高度。

“对齐”：指定水平线在水平位置上的对齐方式，包括默认、左对齐、居中对齐和右对齐等4个选项。只有当水平线的宽度小于浏览器窗口的宽度时该设置才可用。

“阴影”：选中该复选框，水平线则带有阴影效果。

3.5.2 插入日期

向网页中插入系统当前日期的步骤如下。

（1）在文档窗口中将插入点放置到要插入日期的位置。

（2）选择“插入”|“日期”菜单命令，或单击“插入”面板“常用”选项卡中的“日期”图标。

（3）弹出“插入日期”对话框，在该对话框中，可以分别设置星期格式、日期格式和时间格式。并选中“储存时自动更新”复选框。

（4）单击“确定”按钮即可插入日期。

3.5.3 插入特殊字符

特殊字符是指版权符号、注册商标符号、破折号和英镑符号等，这些特殊字符无法从键盘上直接输入，Dreamweaver在“插入”面板上专门设置了常见的特殊字符按钮，单击这些按钮即可完成特殊字符的插入。其具体操作步骤如下。

（1）将光标放置在文档中需要插入特殊字符的位置。

（2）选择“插入”|HTML|“特殊字符”菜单命令，即可插入所需要的字符。如果在“特殊字符”子菜单中没有需要的字符，可以选择“插入”|HTML|“特殊字符”|“其他字符”菜单命令，从中选择需要的特殊字符。

（3）单击需要的字符，单击“确定”按钮，该字符即可插入到文档中。

3.5.4 插入多媒体对象

1. 在网页中插入视频对象

(1) 将光标放置在文档中将要插入视频对象的位置上。

(2) 单击“常用”面板中的“媒体”按钮，这个列表中列出了所有可以在 Dreamweaver 中插入的媒体类型，如图 3.32 所示。下面以插入一个 Flash 为例。



图 3.32 插入多媒体文件

(3) 选择列表中的 SWF 选项，弹出一个“选择文件”对话框，在本地计算机中选择一个 SWF 文件后，单击“确定”按钮即可插入一个 Flash 动画。

(4) 选定插入的 Flash 动画，选择“窗口”|“属性”菜单命令，打开“属性”面板，如图 3.33 所示，这里可以对插入的 Flash 进行以下设置。



图 3.33 Flash“属性”对话框

“宽”和“高”：设置插入的 Flash 动画的宽度和高度。

编辑：单击该按钮，会调出外部编辑器编辑 Flash 文件。

重设大小：单击该按钮，可恢复 Flash 动画的原始尺寸。

“垂直边距”和“水平边距”：设置 Flash 动画的上下和左右边距。

“品质”：用于设置 Flash 动画的质量参数，有低品质、自动低品质、自动高品质和高品质等 4 个选项，选取原则是，品质越低速度越快，制作者根据自身需要选取。

“比例”：用于设置缩放比例。

“对齐”：设置 Flash 动画在网页中的对齐方式。

Wmode：可设置 Flash 动画的透明背景。

“背景颜色”：设置 Flash 动画区域的背景颜色。

“播放”：在设计视图中可以预览 Flash 动画的内容。

“参数”：在参数对话框中可以设定附加参数。

2. 在网页中插入声音

在 Dreamweaver 中可以插入的声音文件有 mp3、wav、midi、aif、ra 和 ram 等。其步骤如下。

(1) 将光标放置在文档中将要插入音频对象的位置上。

(2) 选择“常用”|“媒体”中的“插件”子命令。在本地计算机中选择一个声音文件，然后单击“确定”按钮即可将其添加到页面中。

（3）在文档窗口中只是显示了一个很小的插件图标，为了使它的播放器能在预览时全部显示出来，需要在 Dreamweaver 中将其长和宽都拖拉成合适大小。

（4）在浏览器中预览插入的声音文件。

3.6　表单

表单在网页中表现最多的就是一些用户信息的填写，例如，在申请某个网站的会员时，就需要填写一些个人资料，而实现这些个人资料的功能，都是由 Dreamweaver 中的表单来实现的，表单可以向后台数据库提交这些数据。

3.6.1　表单介绍

表单一般情况下由两部分组成，一部分是描述表单元素的 HTML 源代码，另一部分是客户端处理用户所填信息的程序，如 ASP 和 CGI 等。当浏览者在网页中填写完需要的信息后，单击“提交”按钮，那么输入在表单中的信息就会上传到服务器中，然后再由服务器的有关应用程序进行处理。处理后要么将提交的信息存储在服务器端的数据库中（如银行系统的用户资料），要么将有关的信息返回到客户端的浏览器上（如提示申请会员成功或该用户名重复等）。

3.6.2　表单的使用

表单的创建对象集中在“插入”面板的“表单”选项卡中，如图 3.34 所示。

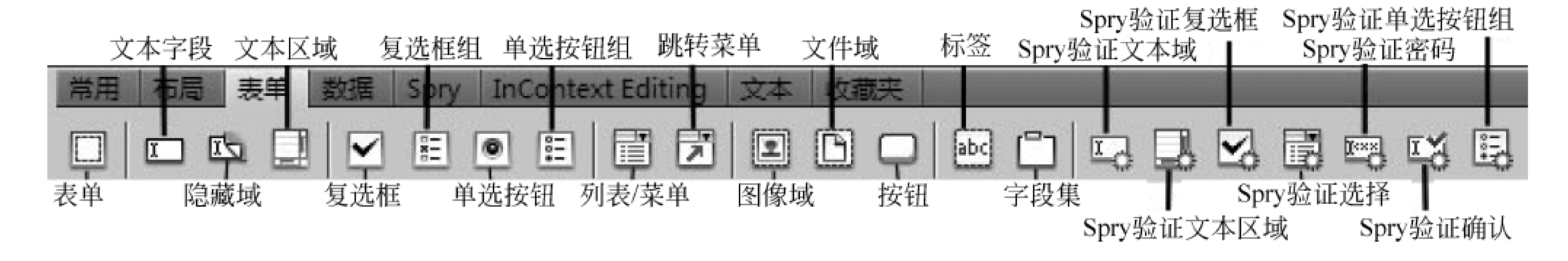


图 3.34　“表单”选项卡

表单是独立的单元，它不能像表格一样任意嵌套。

1. 创建表单域

每一个表单中都包括一个表单域和若干个表单元素。所有的表单元素都要放在这个表单域中才能生效，因此，创建表单时要先插入表单域。其步骤如下。

（1）将光标放置在将要插入表单的位置。

（2）选择“插入”面板中的“表单”选项卡，单击“表单”按钮。

（3）页面上会出现一条红色的虚线，即可插入表单。

（4）选中表单，或在标签选择器中选择＜form＃form1＞标签，可打开表单的“属性”面板，其中可进行以下设置。

①“表单 ID”：输入唯一名称以标识表单。

② “动作”：设置处理该表单动态页或脚本的路径。

③ “方法”：选择表单数据传输到服务器的方法，有以下三种。

- 默认：使用浏览器的默认设置将表单数据发送到服务器。通常的默认方法为 GET。
- GET：将值追加到请求该页的 URL 中。使用 GET 方法发送长表单时，URL 的长度限制在 8192 个字符以内。发送的数据量太大，数据将被截断，从而导致意外或失败的处理结果。由于使用 GET 方法传递信息不安全，因此不用于发送机密用户名和密码、信用卡卡号或其他机密信息。
- POST：在 HTTP 请求中嵌入表单数据。

④ “编码类型”：指定对提交给服务器进行处理的数据使用 MIME 编码类型。

表单域完成后，就可以向其中添加所需要的表单元素了，下面就常见的表单元素的插入及其属性的设置做一介绍。

2. 使用文本域

文本域是可以输入文本内容的表单对象。用户可以创建一个包含单行或多行的文本域，也可以创建一个隐藏用户输入文本的密码文本域。

(1) 单击插入栏中“表单”选项卡中的“文本区域”按钮，一个文本域随即出现在文档中，如图 3.35 所示。

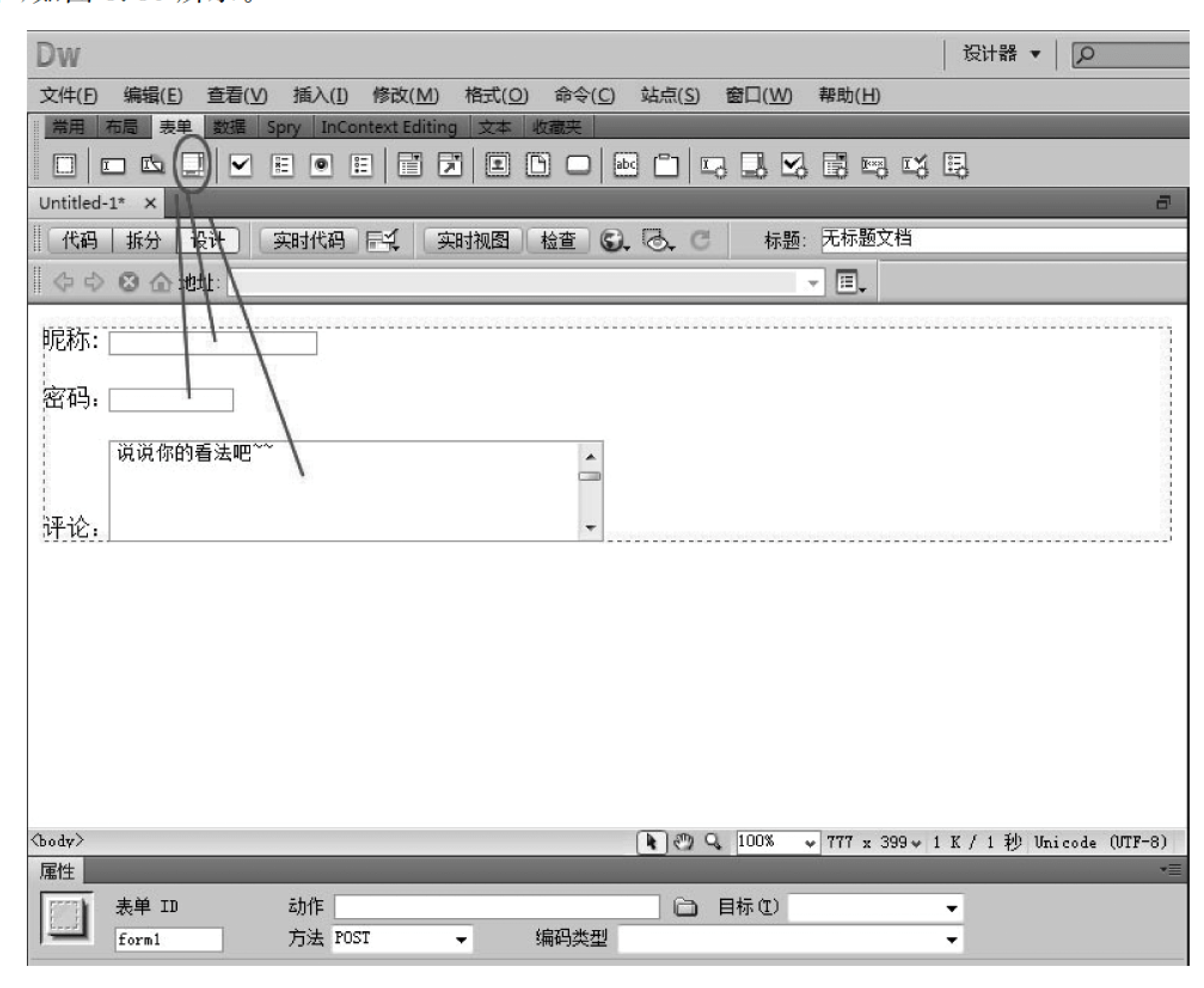


图 3.35 插入文本域

(2) 在属性检查器中,根据需要设置文本域的属性,如图 3.36 所示。



图 3.36　文本域属性设置

“字符宽度”:设置域中可显示的最大字符数。此数值可以小于“最多字符数”。

“最多字符数”:设置单行文本域中可输入的最大字符数,该文本框只会在用户选择“单行”或“密码”选项时会出现。例如,使用“最多字符数”将手机号限制为 11 位数,将密码限制为 10 个字符等。

“类型”:指定是单行、多行或密码域。

① “单行”:输入的文字不会发生换行,一般情况下用于一些简单的输入设置。

② “多行”:输入的文字如果超出了字符宽度,就会产生换行。常见于一些论坛帖子的评论设置。

③ “密码”:产生一个密码文本域。当用户在密码文本域中输入时,输入的内容显示为项目符号或星号,以保护它不被其他人看到。

“行数”:在选中了“多行”单选按钮时可用。设置多行文本域的域高度,实际上就是设置当前这个文本域的高度。

“初始值”:指定在首次载入表单时域中所显示的文字。例如,通过包含说明或示例文字,可以指示用户在域中输入信息。

“类”:将 CSS 应用于对象。

3. 使用复选框

复选框常被用于有多种条件供选择的情况下,用户在填写到这一项时可以选择很多项。每个复选框都是独立的,必须有一个唯一的名称。

(1) 单击插入栏中“表单”选项卡中的“复选框”按钮,一个复选框随即出现在文档中,反复单击“复选框”按钮,可以创建更多的复选框,如图 3.37 所示。

(2) 在属性检查器中,根据需要设置复选框的属性,如图 3.38 所示。

“复选框名称”:为该对象指定一个名称,名称必须在该表单内唯一标识该复选框,且名称不能包含空格或特殊字符。

“选定值”:设置在该复选框被选中时发送给服务器的值。

“初始状态”:确定在浏览器中载入表单时该复选框是否显示被选中状态。

“类”:将 CSS 应用于对象。

4. 使用单选按钮

单选按钮与复选框相似,主要用于标记一个选项是否被选中,单选按钮只允许用户从选项中选择唯一答案。单选按钮通常成组使用,同一组中的所有单选按钮必须具有相同的名

图 3.37 插入复选框



图 3.38 复选框属性设置

称,但它们的域值必须是不同的。

(1) 单击插入栏中"表单"选项卡中的"单选按钮" 插入单选按钮,随即在文档中出现一个单选按钮。反复单击单选按钮,可以创建更多单选按钮,如图 3.39 所示。

(2) 在属性检查器中,根据需要设置单选按钮的属性,如图 3.40 所示。

"单选按钮":为该对象指定一个名称。注意,一组单选按钮必须使用同一个名称,但"选定值"不能相同。

(3) 在浏览器中预览,使用了单选按钮后,一次只能选择一项。

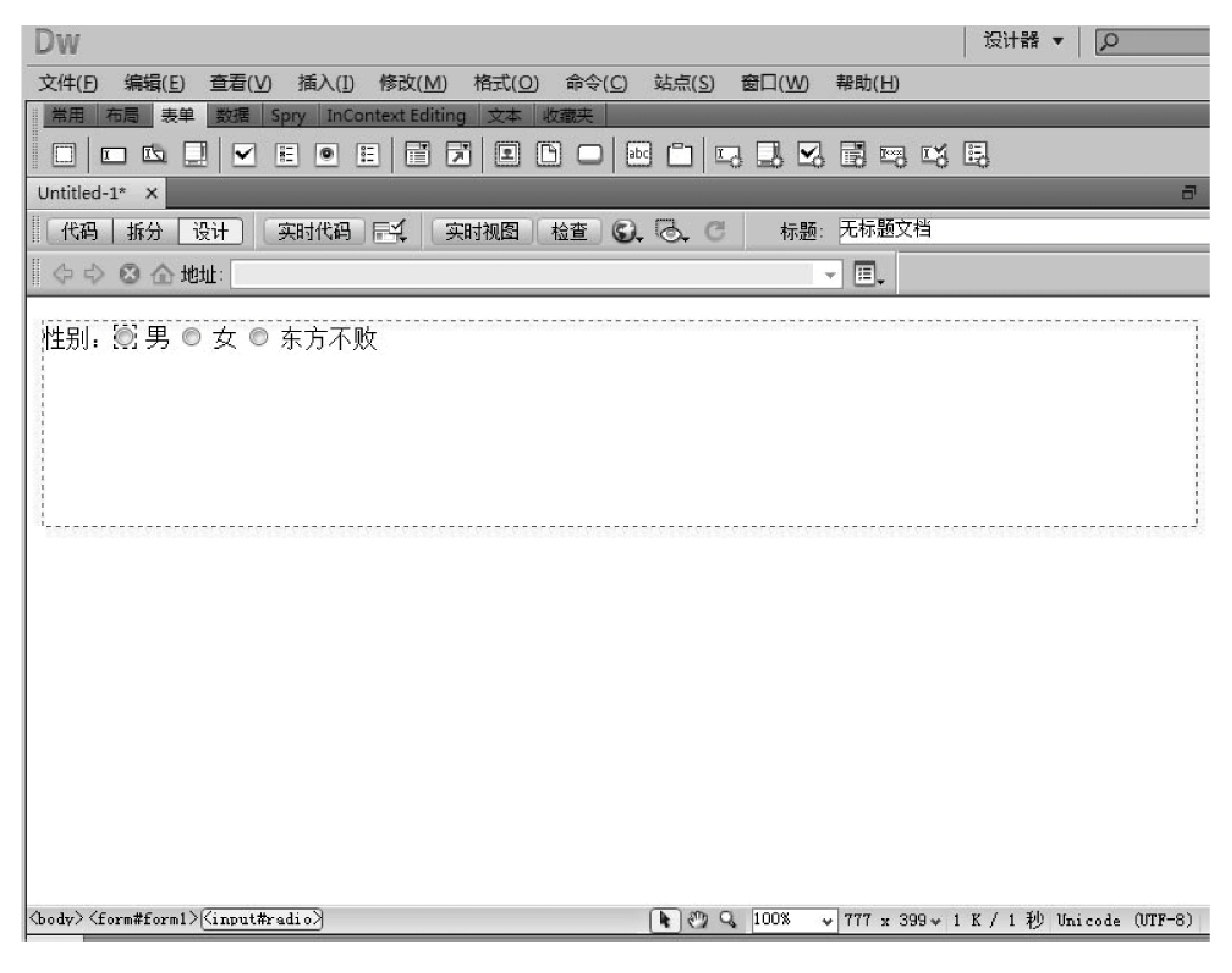


图 3.39　插入单选按钮



图 3.40　单选按钮属性

5. 使用单选按钮组

用户可以用单选按钮一个个地创建表单选项，也可以用单选按钮组来同时创建多个表单选项。在创建多个选项时，单选按钮组比单选按钮的操作更快捷。

（1）将插入点放在表单域内。单击插入栏中“表单”选项卡中的“单选按钮组”按钮。出现“单选按钮组”对话框，如图 3.41 所示。

（2）在这个对话框中可以设置一个单选按钮组中需要包含多少个单选按钮，并且还能够通过标签的设置为每个选项添加内容。

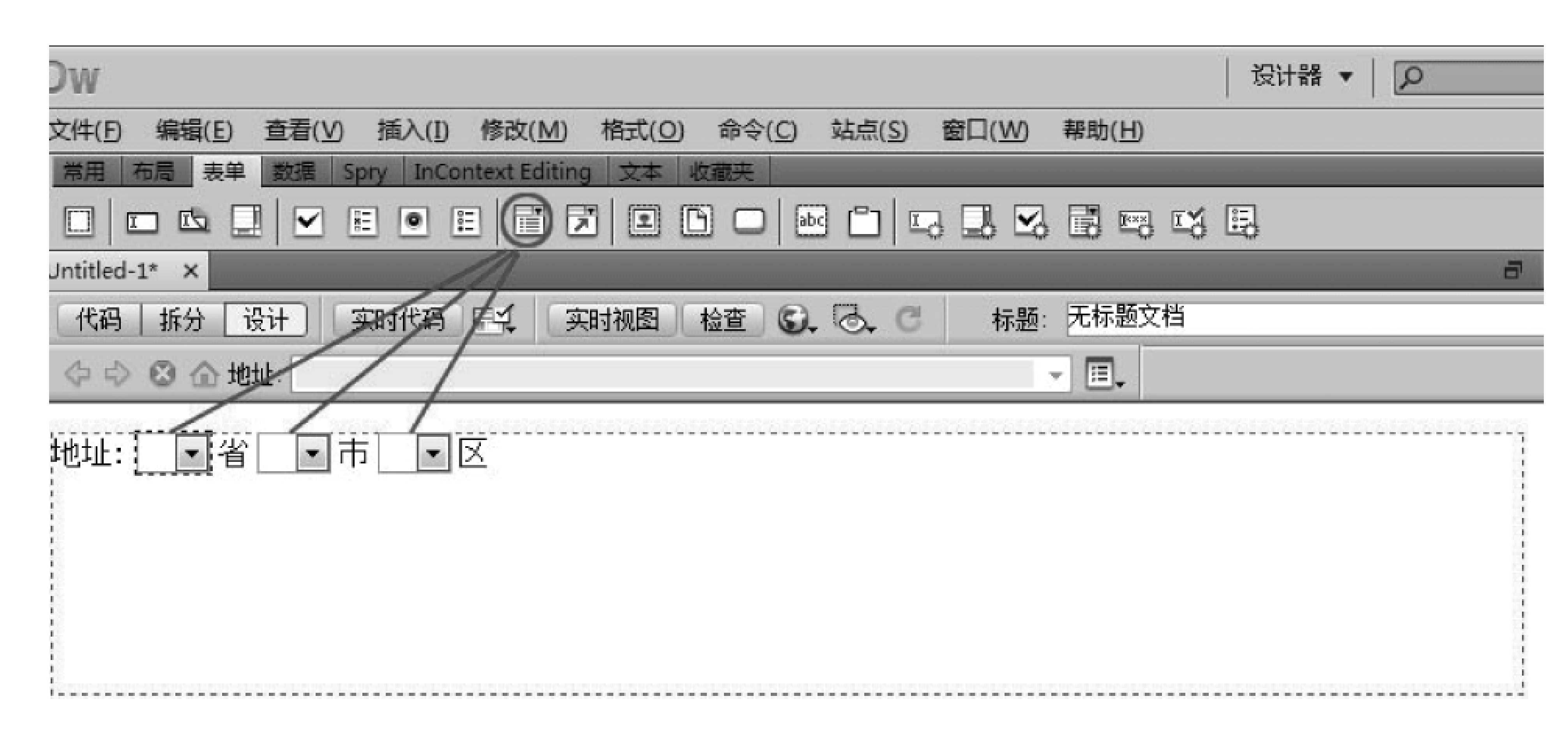
6. 使用列表/菜单

除了选用复选框和单选按钮为用户提供选项，还可以用列表/菜单来制作选项。列表/菜单可以显示一个列有项目的可滚动列表，用户可以从该列表中选择项目。当用户的空间有限又需要显示许多菜单项时，列表/菜单就会非常有用。



图 3.41　插入单选按钮组

(1) 单击插入栏中“表单”选项卡中的“列表/菜单”按钮，文档中便会出现一个菜单，如图 3.42 所示。



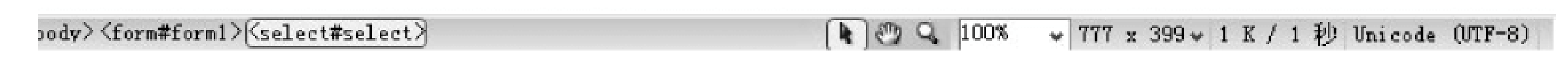


图 3.42　插入列表/菜单

(2) 在属性检查器中根据需要设置其属性。在图 3.42 中需要选择的是地址，所以在属性检查器中将“类型”项选择为“列表”，如图 3.43 所示。

(3) 单击属性检查器中的“列表值”按钮，打开“列表值”对话框。在这里对准备用来选择的项目进行编辑。单击“添加”按钮可以为“项目标签”栏添加列表项，而每个列表项又可

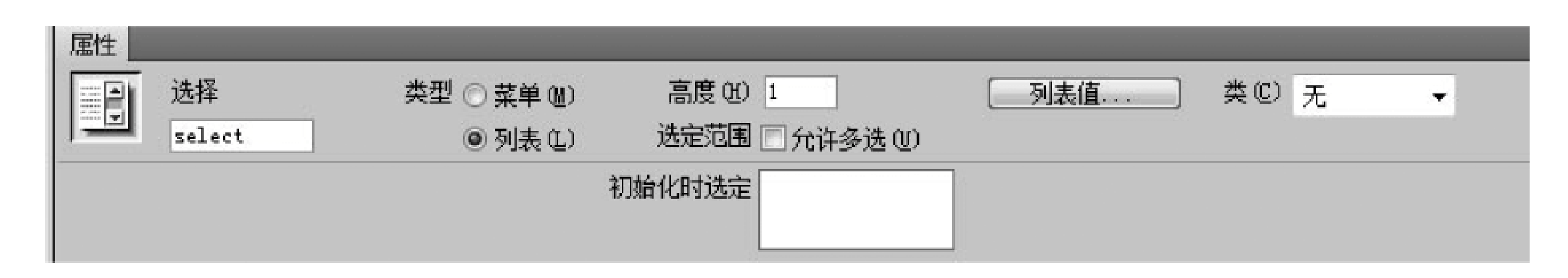


图 3.43 选择“列表”

以通过“值”来设置链接地址，如图 3.44 所示。



图 3.44 添加列表值

(4) 设置完毕后，就可以在浏览器中预览，并对设置好的项目进行选择，如图 3.45 所示。

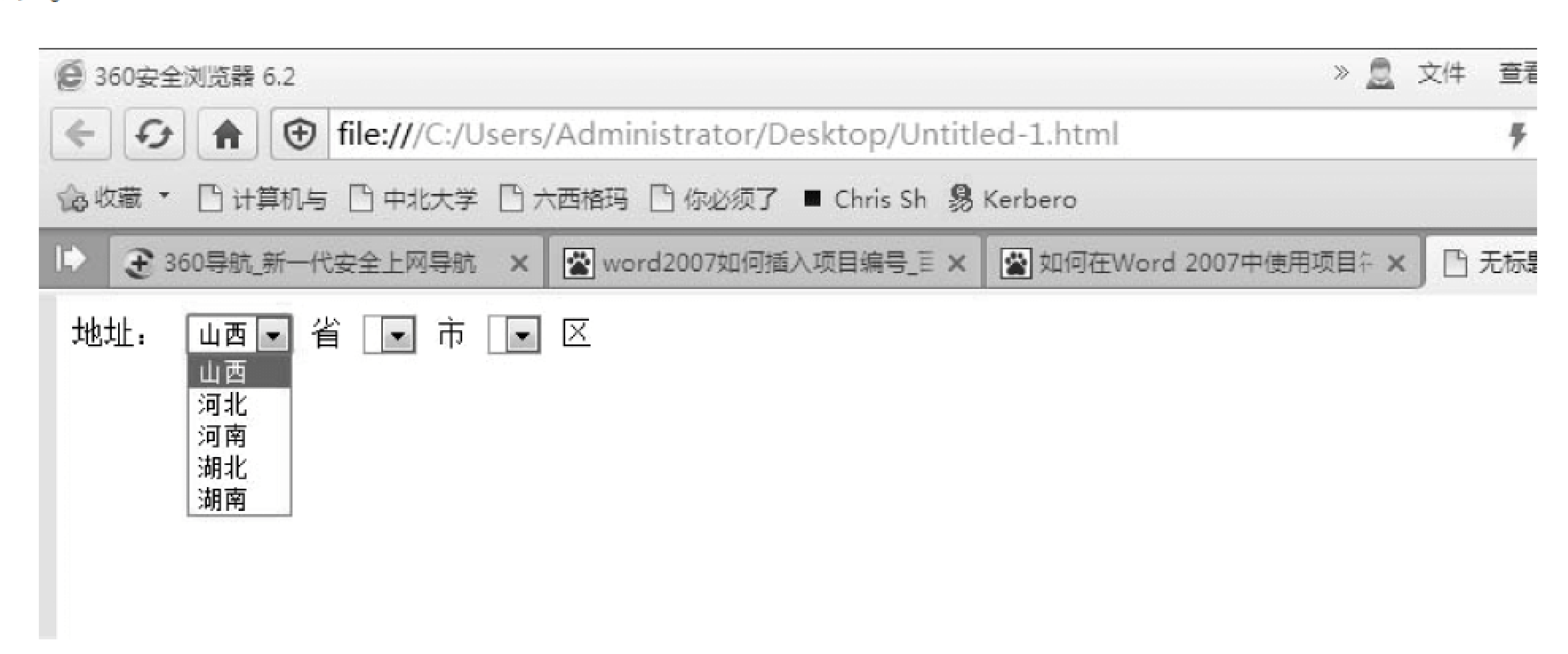


图 3.45 预览效果

7. 使用跳转菜单

跳转菜单利用表单元素形成各种选项的列表。当选中列表中的某个选项时，浏览器会立即跳转到一个新的网页。

(1) 单击插入栏中“表单”选项卡中的“跳转菜单”按钮，随即出现“插入跳转菜单”对话框，根据需要设置其属性。如果在设置的时候，将“菜单之后插入前往按钮”复选框选中，那么插入的跳转表单中就会自动带一个“前往”按钮，如图 3.46 所示。

(2) 设置完毕后可以在浏览器中预览设置好的选项，单击“前往”按钮就可以把当前选中的网页打开。



图 3.46　插入跳转菜单

8. 使用图像域

在网页中插入图像域能够使页面更加丰富多彩。如果使用图像来执行任务而不是提交数据,则需要将某种行为附加到表单对象。

单击插入栏中“表单”选项卡中的“图像域”按钮,即会出现“图像源文件”对话框,选择一张本地计算机上的图片作为表单对象。

如果需要对插入的图像进行编辑,可以在属性检查器中单击“编辑图像”按钮,就可以直接打开 Firework 或 Photoshop 图像处理软件。

9. 使用文件域

文件域对象比文件框对象多了一个“浏览”按钮。浏览者可以通过这个按钮来选择需要上传文件的路径和名称。

(1) 单击插入栏中“表单”选项卡中的“文件域”按钮,随即插入一个文件域对象,如图 3.47 所示。

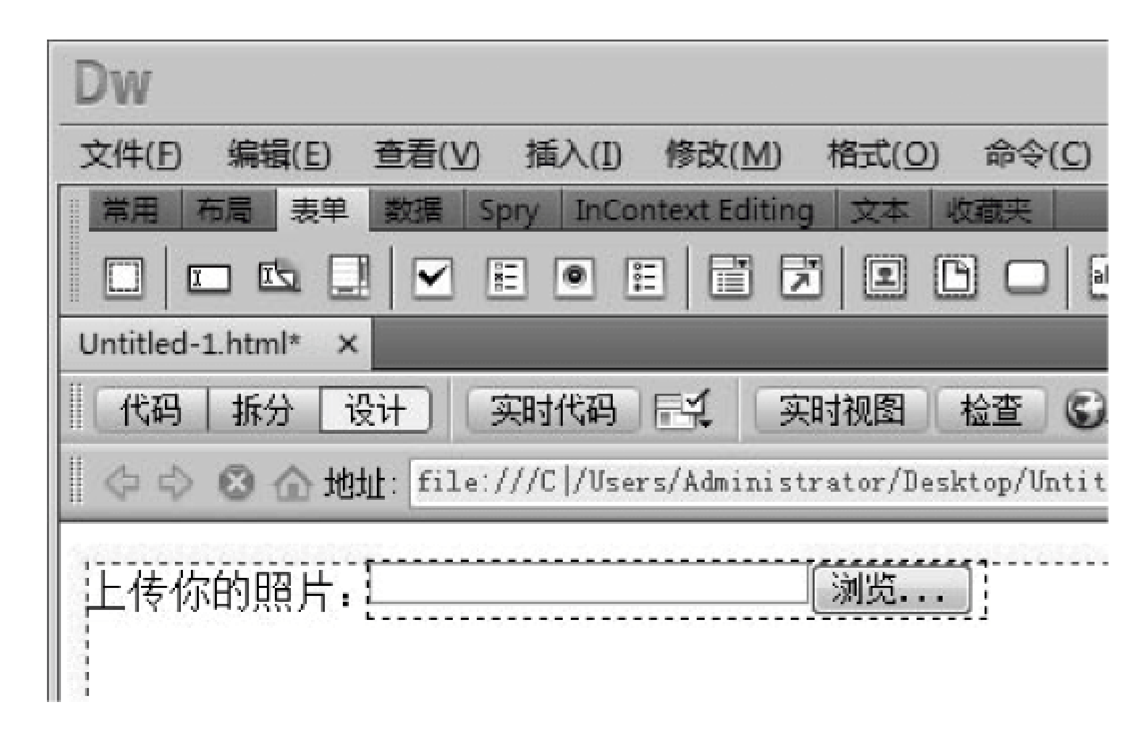


图 3.47　插入文件域

(2) 选中插入的文件域对象，在属性检查器中根据需要可以设置“字符宽度”和“最多字符数”以限制文件域的显示，如图 3.48 所示。

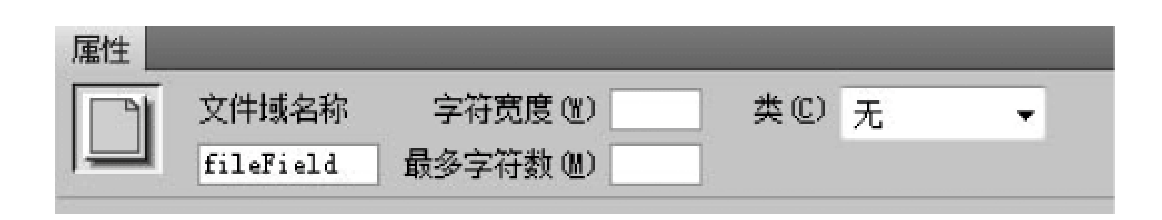


图 3.48 文件域属性

10. 使用标准按钮

按钮对于表单而言必不可少，它可以控制表单的操作。使用按钮可将表单数据提交到服务器，或者重置该表单。在 Dreamweaver 中可以通过图片来制作按钮，也可以使用表单里的按钮。表单中的按钮外形无法改变，但是却比图像按钮用起来更加方便。

(1) 单击插入栏中“表单”选项卡中的 按钮，一个按钮随即出现在文档中，如图 3.49 所示。

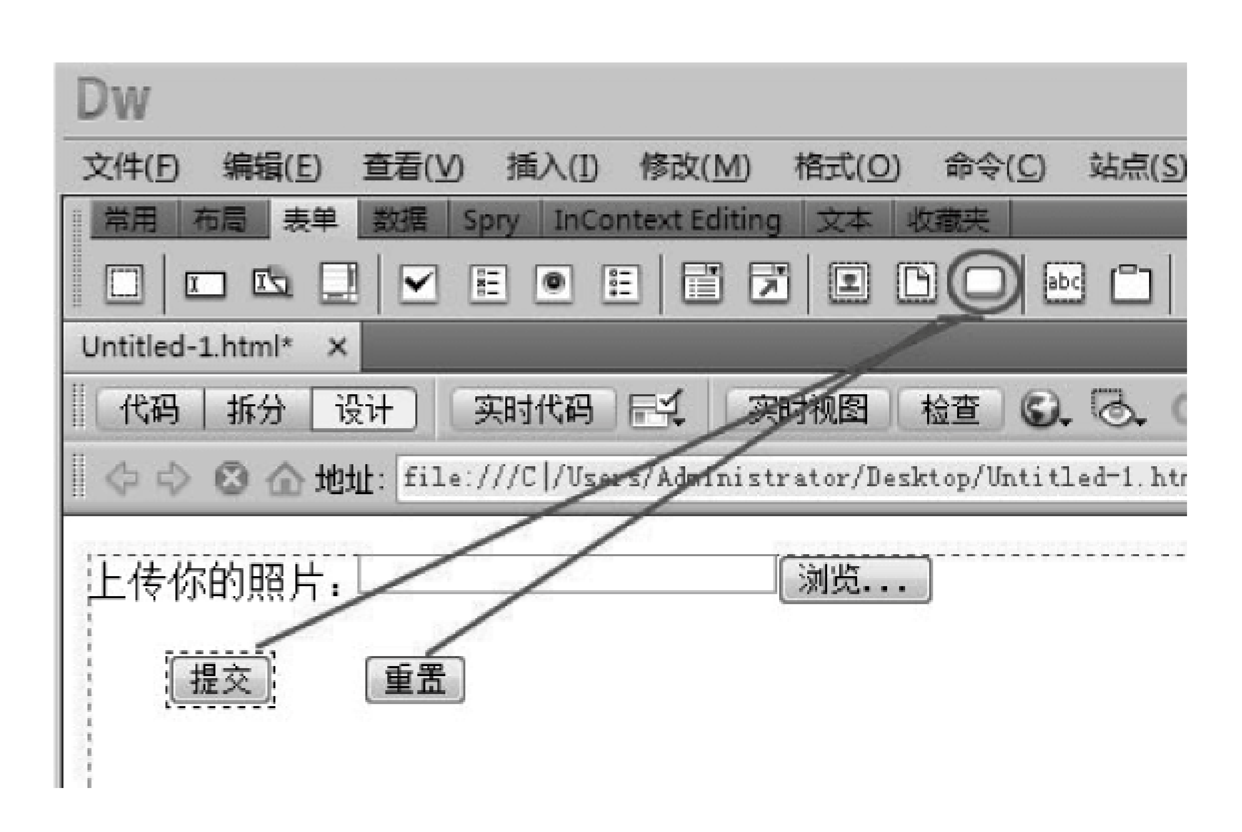


图 3.49 插入按钮

(2) 在属性检查器中根据需要设置该按钮的属性。在“值”项中，可以为这个按钮添加名称，然后再按照名称选择合适的动作，如图 3.50 所示。



图 3.50 设置按钮属性

小结

Dreamweaver 是目前最优秀的网页编辑和网站管理软件之一。本章循序渐进地介绍了 Dreamweaver CS6 的基本操作方法，如输入与编辑文本、建立站点的方法、插入与编辑图像、表格、多媒体元素、表单等。

习题

1. Dreamweaver CS6 的工作区主要包括哪些内容?

2. 常用功能面板主要包括哪些内容?

3. 什么是站点?

4. 在本地计算机中创建站点名称为 my web 的站点,并在该站点中创建名为 image 和 style 的文件夹;在站点内部创建 index. html,jieshao. html,yinyue. html,shiyin. html,zhuce. html 和 div. css 文件。向 image 文件夹内复制多张图像,练习复制、移动、删除、粘贴等操作。

5. 在第 4 题中创建的 index. html 中,输入文字,插入图像、多媒体元素等练习相关编辑操作,创建各种超链接。

第4章 CSS样式网页布局规划

本章要点：

- CSS3 介绍；
- CSS3 基础语法；
- 在 Dreamweaver 中使用 CSS 样式；
- 编辑 CSS 样式；
- 网页布局规划。

一个网页设计者对于 HTML 一定不陌生，因为它是所有网页制作的基础。但是如果希望网页能够美观、大方，并且升级方便，维护轻松，那么仅知道 HTML 是不够的，CSS 在这中间扮演着重要的角色。本章从 CSS 的基础概念出发，介绍 CSS 语言的特点，以及如何在网页中应用 CSS，并对常见网页类型如何用 CSS 进行布局进行简要介绍。

4.1 CSS3 介绍

4.1.1 CSS3 功能

CSS(Cascading Style Sheets)称为层叠样式表，也称为 CSS 样式表或样式表，其文件扩展名为.css。CSS 是用于增强或控制网页样式，并将样式信息与网页内容分离的一种标记语言。

引用样式表的目的是将“网页结构代码”和“网页样式风格代码”分离开，从而使网页设计者可以对网页布局进行更多的控制。利用样式表，可以将整个网站上所有的网页都指向某个 CSS 文件，设计者只需要修改 CSS 文件中的某一行，整个网页上对应的样式都会随之发生改变。

CSS 目前有三个不同层次的标准，即 CSS1，CSS2，CSS3。

CSS1 主要定义了网页的基本属性，如字体、颜色、空白边等。CSS2 在此基础上添加了一些高级功能，如浮动和定位，以及一些高级选择器，如子选择器、相邻选择器和通用选择器等。CSS3 开始遵循模块化开发，这有助于理清模块化规范之间的不同关系，减少完整文件的大小。以前的规范是一个完整的模块，实在是太庞大、太复杂，所以，新的 CSS3 规范将其分成了多个模块。

CSS3 制定完成后，具有了很多新功能，即新样式。但这些新样式在浏览器中不能获得完全支持，主要在于各种浏览器对 CSS3 很多细节处理上存在差异。但 CSS3 是一个具有强大发展潜力的新技术，在样式修饰方面，它具有不可取代的地位。

4.1.2 CSS 规则

1. CSS 基本语法

CSS 文件是纯文本格式文件，可以使用系统自带的文本工具编辑，如常见的记事本；也可以选择专业的 CSS 编辑工具，如 Dreamweaver。

CSS 样式表由若干条样式规则组成，这些样式规则可以应用到不同的元素或文档来定义它们显示的外观。每一条样式规则由三部分构成：选择符(selector)、属性(properties)和属性值(value)，基本格式如下：

```
selector{property:value}
```

(1) selector 选择符有多种形式，既可以是文档中的 HTML 标记，如<body>、<table>、<p>等，也可以是 XML 文档中的标记。

(2) property 属性则是选择符指定的标记所包含的属性。

(3) value 指定了属性的值。如果定义选择符的多个属性，则属性和属性值为一组，组与组之间用分号(;)隔开。基本格式如下：

```
selector{property1:value;property2:value2;…}
```

下面给出一条样式规则，如：

```
p{color:red}
```

该样式规则的选择符为 p，为段落标记<p>提供样式，color 为指定文字颜色属性，red 为属性值。此样式表示标记<p>指定的段落文字为红色。

如果要为段落设置多种样式，则可以使用下列语句：

```
p{font-family:Tahoma, Geneva, sans-serif;color:red;font-size:12px;font-weight:bold}
```

2. 手工编写 CSS

CSS 是文本格式，传统的文本编辑器(如记事本或 Word)都可以编辑 CSS，但这些编辑工具软件不支持语法提示，不支持验证，严重影响开发效率。但使用记事本手工编写 CSS，有利于初学者更快地掌握 CSS3 技术。

使用记事本编写 CSS 和使用记事本编写 HTML 文档基本一样。首先需要打开一个记事本，然后在里面输入相应的 CSS 代码即可，具体步骤如下。

(1) 打开记事本，编写 HTML 文档，如图 4.1 所示。

(2) 添加 CSS 代码，修饰 HTML 元素。在 head 标记中间，添加 CSS 样式代码，如图 4.2 所示。



图 4.1 用记事本编写 HTML 文档



图 4.2 添加 CSS 代码

从窗口中可以看出，在 head 标记中间，添加了一个 style 标记，即 CSS 样式标记。在 style 标记中间，对 p 进行了设定，这里设置段落居中显示并且颜色为红色。

(3) 网页编辑完后，保存为扩展名为.html 的文件，使用浏览器打开，如图 4.3 所示，可以看到段落在网页中间以红色字体显示。

3. Dreamweaver 编写 CSS

Dreamweaver 编辑器有语法着色，带输入提示，甚至有自动创建 CSS 的功能，深受开发人员喜爱。使用 Dreamweaver 创建 CSS 步骤如下。

(1) 使用 Dreamweaver 创建 HTML 文档，如图 4.4 所示。

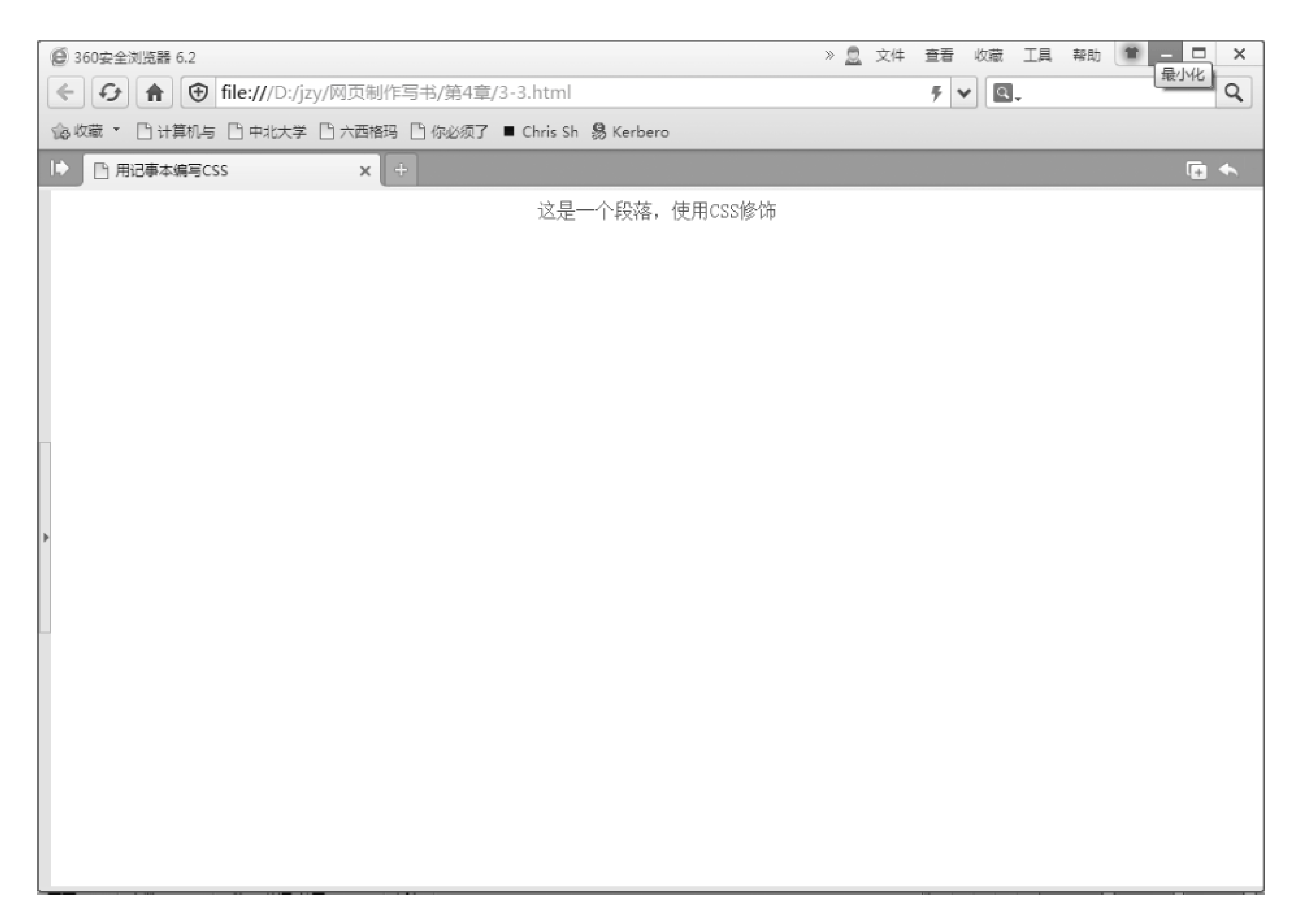


图 4.3　浏览器预览效果

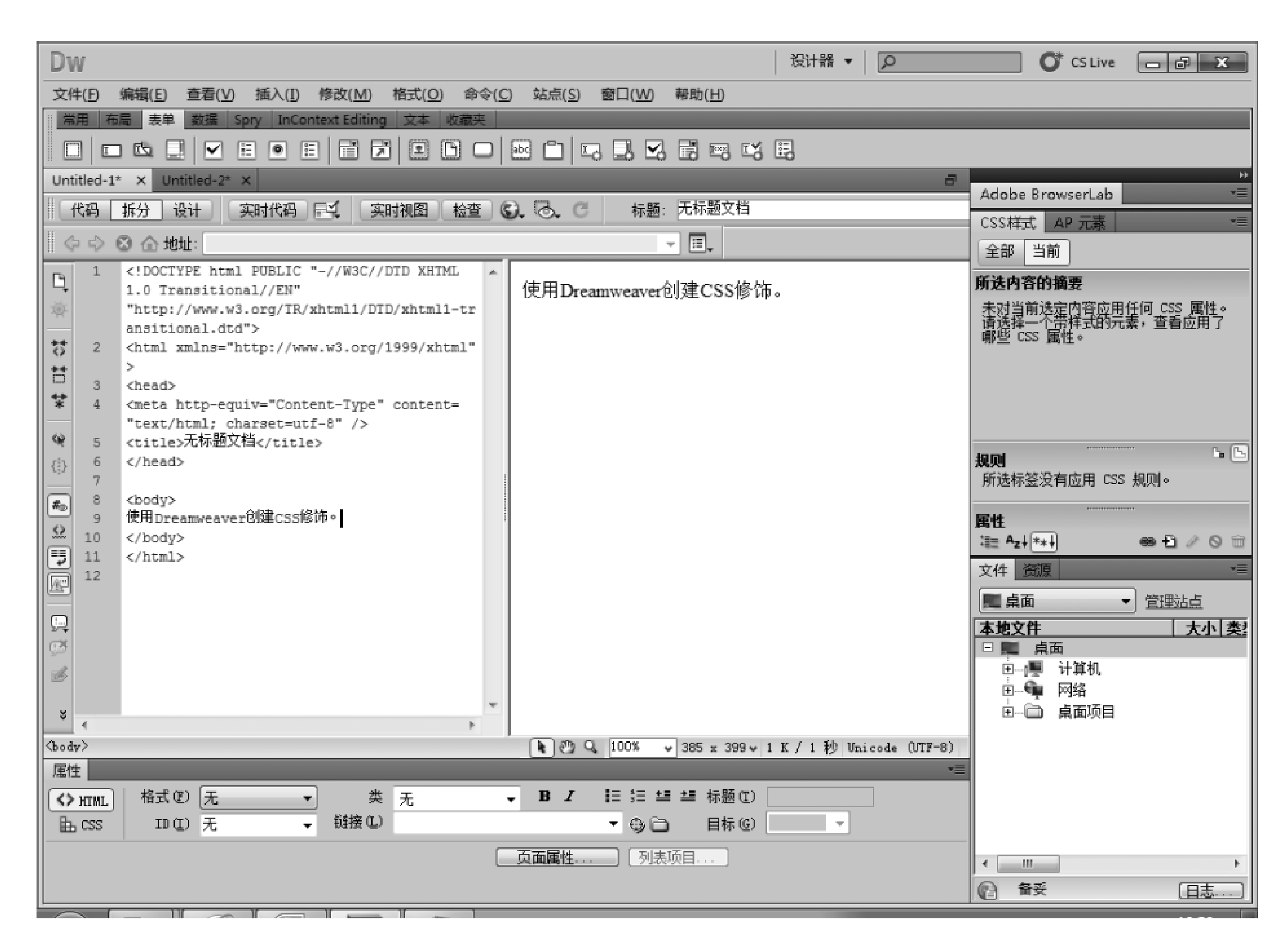


图 4.4　创建 HTML 文档



(2) 在设计模式中，选中需创建 CSS 的文本，右击并在弹出的快捷菜单中选择“CSS 样式”|“新建”命令，弹出如图 4.5 所示的对话框。

图 4.5　“新建 CSS 规则”对话框

该对话框中，在“为 CSS 规则选择上下文选择器类型”下拉列表中，选择“标签(重新定义 HTML 元素)”选项(后续内容学过后，读者可根据需要选择不同的选择器类型)。选择完成后，单击“确定”按钮，会显示如图 4.6 所示的对话框。

图 4.6　选择类型

该对话框中，可以选择 p 的各种样式设置。完成选择后，单击“确定”按钮，就完成 p 样式的设置了。效果如图 4.7 所示。从代码窗口来看，可以看到在<head>标记中，增加了一个 style 标记，用来放置 CSS 样式。其样式用来修饰段落 p。

除了上述使用 Dreamweaver 设置 CSS 外，用户还可以直接在代码视图中编写 CSS 代码，此时会有语法提示。

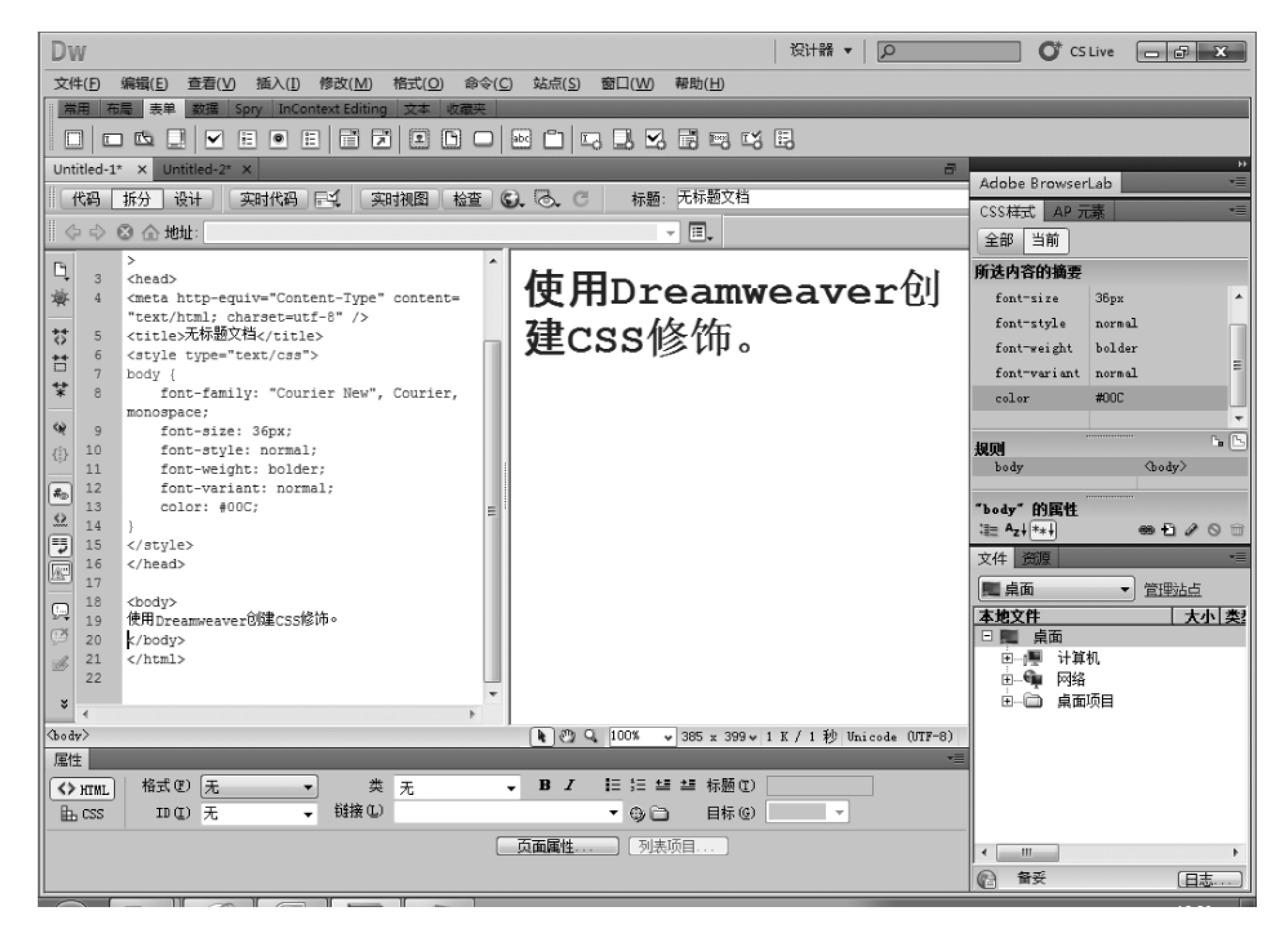


图 4.7 CSS 设置效果

4.1.3 链接和导入 CSS 样式

CSS 样式表能很好地控制页面显示，以达到分离网页内容和样式代码。CSS 样式表控制 HTML5 页面达到好的样式效果，其方法通常包括行内样式、内嵌样式、链接样式和导入样式。

1. 行内样式

行内样式是所有样式中比较简单、直观的方法，就是直接把 CSS 代码添加到 HTML 的标记中，即作为 HTML 标记的属性标记存在。通过这种方法，可以很简单地对某个元素单独定义样式。

使用行内样式方法是直接在 HTML 标记中使用 style 属性，该属性的内容就是 CSS 的属性和值。例如：

```
<p style="font-size:18px;">行内样式</p>
```

【例 4.1】 行内样式实例。

```
<html>
<head>
<title>行内样式</title>
</head>
<body>
```

```
    <p style="color:red;font-size:20px;text-decoration:underline;
text-align:center">此段落使用行内样式修饰</p>
    <p style="color:blue;font-style:italic">正文内容</p>
    </body>
    </html>
```

网页显示效果如图 4.8 所示。



图 4.8 浏览器显示效果

尽管行内样式简单，但这种方法不常使用。因为这样添加无法发挥样式表“内容结构与样式控制代码”相分离的优势。并且这种方式也不利于样式的重用，如果为每一个标签都设置 style 属性，后期维护成本过高，网页冗余量大，运行缓慢。

2. 内嵌样式

内嵌样式就是将 CSS 样式代码添加到<head>与</head>之间，并且用<style>和</style>标记进行声明。这种写法虽然没有完全实现页面内容和样式控制代码完全分离，但可以设置一些比较简单的样式，并统一页面样式。

其格式如下：

```
<head>
 <style type="text/css">
 <!--
   p
    {
      Color:red;
      Font-size:12px;
```

```
    }
  -->
 </style>
</head>
```

<! --…-->的作用是,如某些低版本的浏览器无法解读这段样式代码,则跳过这段代码不进行解析。这是为了防止低版本的浏览器显示错误。建议加上这个标记。

【例 4.2】 内嵌样式实例。

```
<html>
<head>
<title>内嵌样式</title>
<style type="text/css">
p{
       color:orange;
       text-align:center;
       font-weight:bolder;
       font-size:25px;
}
</style>
</head>
<body>
<p>此段落使用内嵌样式修饰</p>
<p>正文内容</p>
</body>
</html>
```

网页显示效果如图 4.9 所示。



图 4.9 浏览器显示效果

内嵌样式只适用于为特殊页面设置单独的样式风格。

3. 链接样式

链接样式是 CSS 中使用频率最高的方法。它将“页面内容”和“样式风格代码”分离成两个或多个文件，实现了页面框架 HTML 代码和 CSS 代码的完全分离，使前期制作和后期维护都十分方便。同一个 CSS 文件，根据需要可以链接到网站中所有的 HTML 页面上。

链接样式是指在外部定义 CSS 样式表并形成以.css 为扩展名的文件，然后在页面中通过<link>链接标记链接到网页中，该链接语句必须放置在页面的<head>标记区，如下所示：

```
<link rel = "stylesheet" type = "text/css" href = "1.css"/>
```

(1) rel 指定链接到样式表，其值为 stylesheet。
(2) type 表示样式表类型为 CSS 样式表。
(3) href 指定了 CSS 样式表的所在位置，此处表示当前路径下名称为 1.css 的文件。

【例 4.3】 链接样式实例。

(1) 链接样式的.html 文件。

```
<html>
<head>
<title>链接样式</title>
<link rel = "stylesheet" type = "text/css" href = "3.5.css" />
</head>
<body>
<h1 > CSS 学习</h1 >
<p>此段落使用链接样式修饰</p>
</body>
</html>
```

(2) 链接样式的.css 文件。

```
h1{text - align:center;}
p{font - weight:29px;text - align:center;font - style:italic;}
```

其网页中的显示效果如图 4.10 所示。

链接样式的最大优势是将 CSS 代码和 HTML 代码完全分离，并且同一个 CSS 文件能被不同的 HTML 所链接使用。

4. 导入样式

导入样式和链接样式基本相同，都是创建一个单独的 CSS 文件，然后再引入到 HTML 文件中。只不过语法和运作方式有差别。采用导入样式的样式表，在 HTML 文件初始化时，会被导入到 HTML 文件内，作为文件的一部分，类似于内嵌效果。而链接样式是在 HTML 标记需要样式风格时才以链接方式引入。

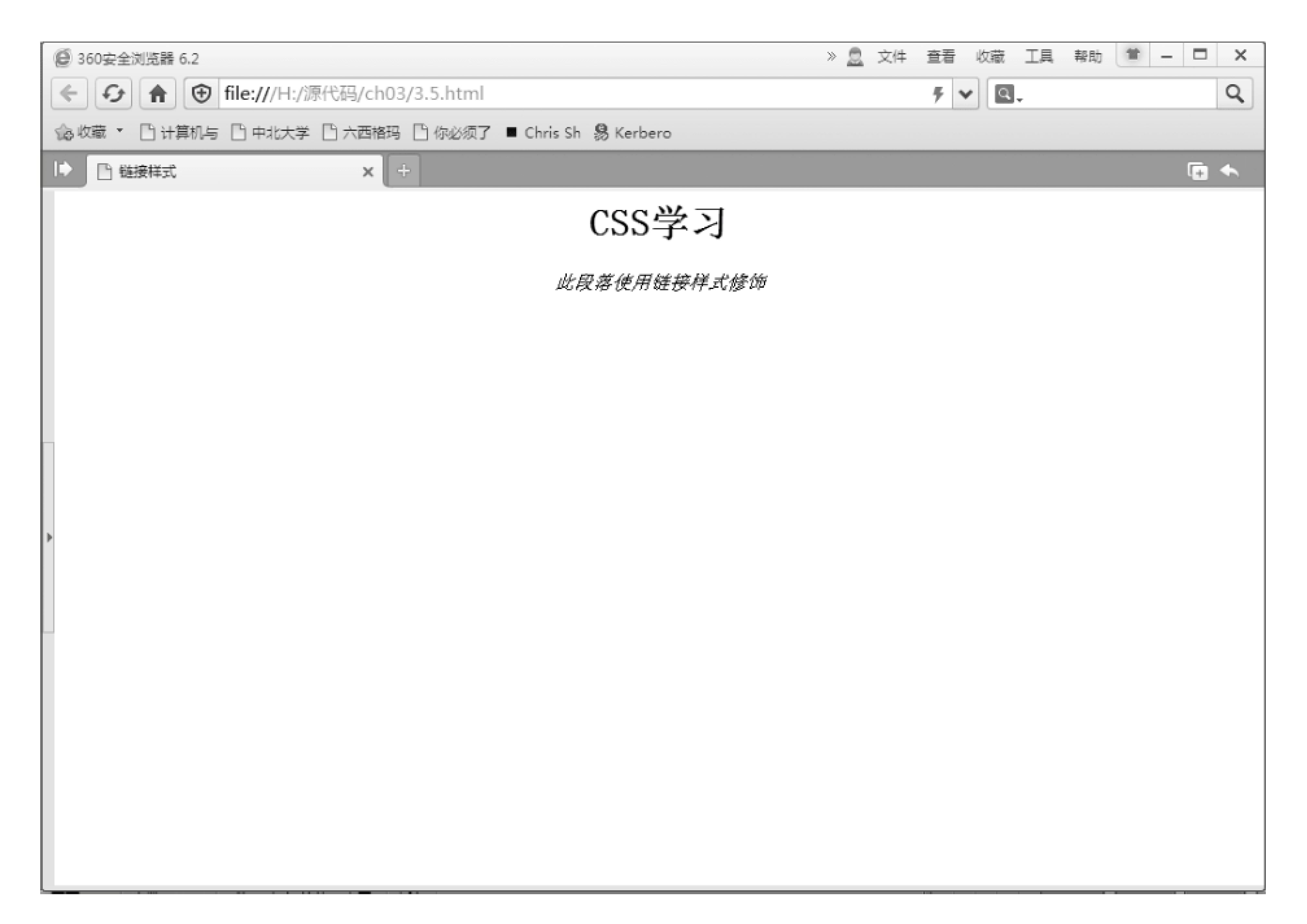


图 4.10 浏览器显示效果

导入外部样式表是指在内部样式表的<style>标记中，使用@import 导入一个外部样式表，例如：

```
<head>
 <style type="text/css">
  <!—
  @import "1.css"
    -->
 </style>
</head>
```

导入外部样式表相当于将样式表导入到内部样式表中，其方法更有优势。导入外部样式表必须在样式表的开头部分，即其他内部样式表前面。

【例 4.4】 导入样式。

(1) .html 文件部分。

```
<html>
<head>
<title>导入样式</title>
<style>
@import "3.6.css"
</style>
</head>
<body>
<h1>CSS 学习</h1>
```

```
<p>此段落使用导入样式修饰</p>
</body>
</html>
```

(2) .css 部分。

```
h1{text-align:center;color:#0000ff}
p{font-weight:bolder;text-decoration:underline;font-size:20px;}
```

网页显示效果如图 4.11 所示。



图 4.11　浏览器显示效果

导入样式与链接样式比较,最大的优点是可以一次导入多个 CSS 文件,其格式如下:

```
<sytle>
@import "p.css"
@import "text.css"
</style>
```

4.1.4　优先级问题

如果同一个页面采用了多种 CSS 使用方式,且这几种样式共同作用于一个标记,就会出现优先级问题,即究竟哪种样式设置有效。

行内优先级大于内嵌优先级;内嵌优先级大于链接优先级;链接优先级大于导入优先级。

4.2 CSS3 基础语法

4.2.1 CSS3 选择器

HTML 中的所有标记都是通过不同的 CSS 选择器(selector)进行控制的。

1. 标签选择器

HTML 文档是由多个不同标记组成,而 CSS 选择器就是声明哪些标记采用样式。例如,p 选择器就是用于声明页面中所有<p>标记的样式风格。

标签选择器的基本形式如下:

```
tagname{property:value}
```

其中,tagname 表示标记名称,如 p、h1 等 HTML 标记;property 表示 CSS3 属性;value 表示属性值。

通过一个具体标记来命名,可以对文档里这个标记出现的每一个地方应用样式定义。这种做法通常用在设置那些在整个网站中都会出现的基本样式。

【例 4.5】 标签选择器实例。

```
<html>
<head>
<title>标签选择器</title>
<style>
p{color:blue;font-size:20px;}
</style>
</head>
<body>
<p>此处使用标签选择器控制段落样式</p>
</body>
</html>
```

在浏览器中的浏览效果如图 4.12 所示。

2. 类选择器

在一个页面中,使用标签选择器,会控制该页面中所有此标记的显示样式。如果需要为此类标记中其中一个标记重新设定,此时仅使用标签选择器是不能达到效果的,需要使用类(class)选择器。

类选择器用来为一系列标记定义相同的呈现方式,常用语法格式如下:

```
.classvalue{property:value}
```

其中,classvalue 是选择器的名称。例如:

```
.rd{color:red}
.se{font-size:3px}
```

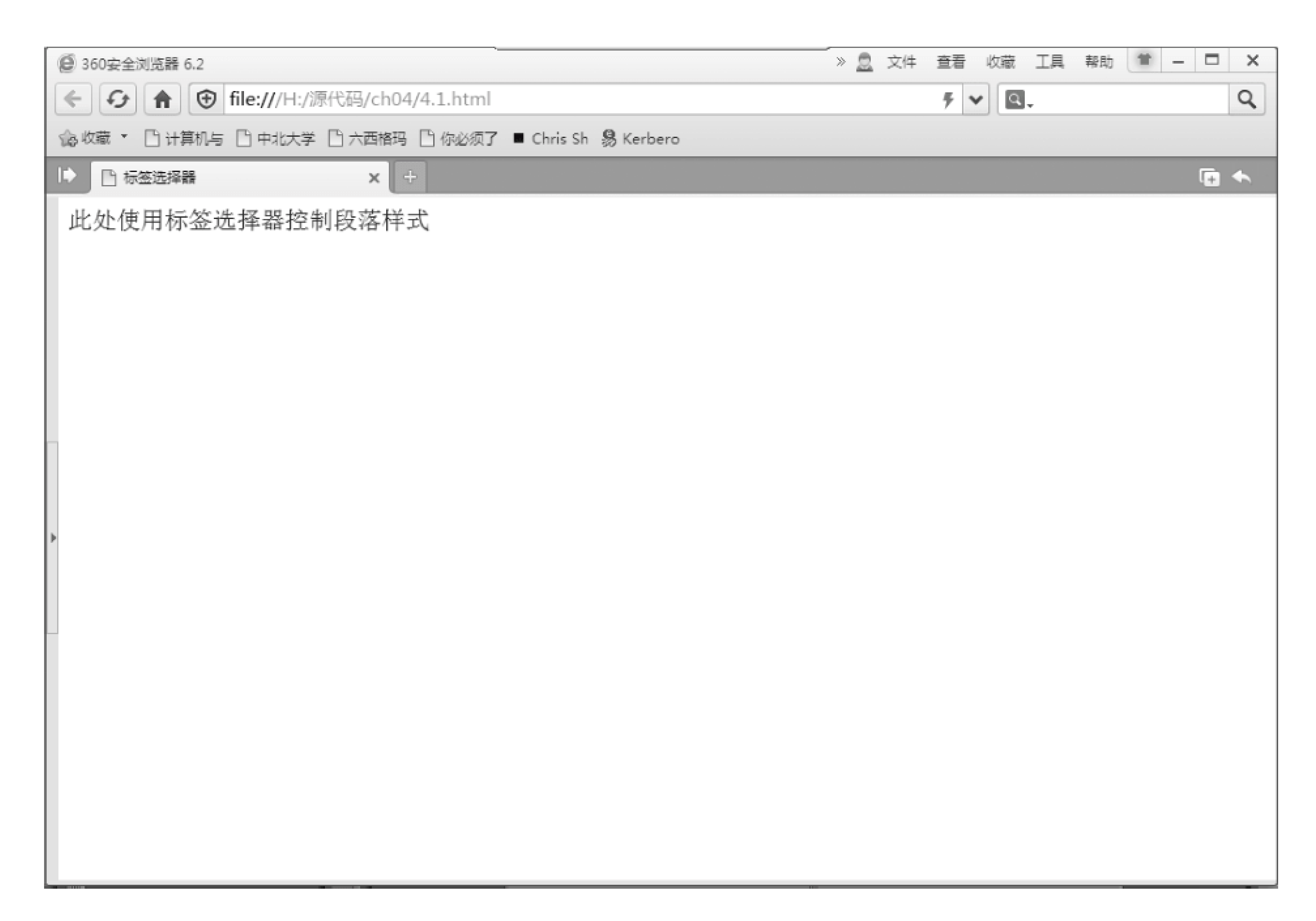


图 4.12 浏览器显示效果

上面定义了两个类选择器，分别为 rd 和 se。类的名称可以是任意英文字符串或以英文开头与数字的组合，一般情况下，是其功能及效果的简写。

可以使用 p 标记的 class 属性来引用类选择符，其使用示例如下：

```
<p class = "rd"> class 属性是被用来引用类选择器的属性.</p>
```

这样，前面定义的选择器就被应用于指定的标记 p 中。如果需要在不同标记中使用相同的呈现方式，如下所示：

```
<p class = "rd">段落样式</p>
<h3 class = "rd">标题样式</h3>
```

【例 4.6】 类选择器实例。

```
<html>
<head>
<title>类选择器</title>
<style>
.aa{
   color:blue;
   font - size:20px;
}
.bb{
    color:red;
    font - size:22px;
}
```

```
</style>
</head>
<body>
<h3 class = bb>学习类选择器</h3>
<p class = "aa">此处使用类选择器 aa 控制段落样式</p>
<p class = "bb">此处使用类选择器 bb 控制段落样式</p>
</body>
</html>
```

在浏览器中显示效果如图 4.13 所示。



图 4.13 浏览器显示效果

3. ID 选择器

ID 选择器和类选择器类似，都是针对特定属性的属性值进行匹配。ID 选择器定义的是某个特定的 HTML 元素，一个网页文件中只能有一个元素使用某一个 ID 的属性值。

其基本语法格式如下：

```
#idvalue{property:value}
```

其中，idvalue 是选择器名称。具体使用如下示例：

```
#fontstyle
{
Color:red;
Font-size:large;
Font-weight:bold
}
```

在页面中，具有 ID 属性的标记才能使用 ID 选择器定义样式，所以与类选择器相比，使用 ID 选择器是有一定局限性的。类选择器与 ID 选择器主要有以下两点区别。

（1）类选择器可以给人以数量的标记定义样式，但 ID 选择器在页面的标记中只能使用一次。虽然在很多浏览器下，ID 选择器用于多个标记时也能正确显示样式，但这种使用方式是错误的。因为每个标记定义的 ID 不只是 CSS 可以调用，JavaScript 等脚本语言同样也可以调用。如果一个 HTML 中有两个相同 ID 的标记，那么将会导致 JavaScript 在查找 ID 时出错。

（2）ID 选择器比类选择器具有更高的优先级。

【例 4.7】 ID 选择器实例。

```
<html>
<head>
<title>ID 选择器</title>
<style>
#fontstyle{
   color:blue;
   font-weight:bold;
}
#textstyle{
    color:red;
    font-size:22px;
}
</style>
</head>
<body>
<h3 id=textstyle>学习 ID 选择器</h3>
<p id=textstyle>此处使用 ID 选择器 aa 控制段落样式</p>
<p id=fontstyle>此处使用 ID 选择器 bb 控制段落样式</p>
</body>
</html>
```

在浏览器中显示效果如图 4.14 所示。

4. 全局选择器

如果想要一个页面中所有 HTML 标记都使用同一种样式，可以使用全局选择器。其语法格式为：

```
*{property:value}
```

其中，* 表示对所有元素起作用。使用示例如下：

```
*{margin:0;padding:0;}
```

【例 4.8】 全局选择器实例。

```
<html>
<head>
<title>全局选择器</title>
```

```
<style>
 * {
   color:red;
   font-size:30px
}
</style>
</head>
<body>
<p>使用全局选择器修饰</p>
<p>第一段</p>
<h1>第一段标题</h1>
</body>
</html>
```



图 4.14 浏览器显示效果

在浏览器中的显示效果如图 4.15 所示。

5. 组合选择器

组合选择器只是一种组合形式,并不算是一种真正的选择器,但在实际中经常使用,其使用示例如下:

```
.orderlist li{ *** }
.tableset td{ *** }
```

在使用时一般用在重复出现并且样式相同的一些标签里,如 li 列表、ID 单元格和 dd 自定义列表等。



图 4.15　浏览器显示效果

【例 4.9】 组合选择器实例。

```
<html>
<head>
<title>组合选择器</title>
<style>
p{
  color:red
}
p.firstPar{
  color:blue
}
.firstPar{
  color:green
}
</style>
</head>
<body>
<p>这是普通段落</p>
<p class = "firstPar">此处使用组合选择器</p>
<h1 class = "firstPar">我是一个标题</h1>
</body>
</html>
```

在浏览器中显示效果如图 4.16 所示。



图 4.16 浏览器显示效果

6. 继承选择器

继承选择器的规则是，子标记在没有定义的情况下所有的样式是继承父标记的，当子标记重复定义了父标记已经定义过的声明时，子标记就执行后面的声明，与父标记不冲突的地方仍然沿用父标记的声明。使用示例如下：

```
<div class = "text">
<span><img src = " * * * " alt = "示例图片"></span>
</div>
```

对于上面层而言，如果其修饰样式为下面的代码：

```
.text span img{border:1px blue solid;}
```

则表示该选择器先找到 class 为 text 的标记，再从它的子标记里查找 span 标记，再从 span 的子标记中找到 img 标记。也可以采用下面的形式：

```
div span img {border:1px blue solid;}
```

【例 4.10】 继承选择器实例。

```
<html>
<head>
<title>继承选择器</title>
<style type = "text/css">
h1{color:red; text - decoration:underline;}
```

```
h1 strong{color:#004400; font-size:40px;}
</style>
</head>
<body>
<h1>测试 CSS 的<strong>继承</strong>效果</h1>
<h1>此处使用继承<font>选择器</font>了么?</h1>
</body>
</html>
```

在浏览器中的效果如图 4.17 所示。



图 4.17　浏览器显示效果

7. 伪类

伪类也是选择器的一种,但用伪类定义的 CSS 样式并不是作用在标记上,而是作用在标记的状态上。伪类包括:first-child、:link、:vistited、:hover、:active、:focus 和:lang 等。

其中有一组伪类是主流浏览器都支持的,就是超级链接标记<a>的伪类,包括::link(未访问链接)、:vistited(已访问链接)、:hover(鼠标停留在链接上)和:active(激活链接)。使用示例如下:

```
a:link{color:#ff0000; text-decoration:none}
a:visited{color:#00ff00; text- decoration:none}
a:hover{color:#0000ff; text- decoration:underline}
a:active{color:#ff00ff; text- decoration:underline}
```

【例 4.11】 伪类实例。

```
<html>
```

```
<head>
<title>伪类</title>
<style>
a:link {color: red}                /* 未访问的链接 */
a:visited {color: green}           /* 已访问的链接 */
a:hover {color:blue}               /* 鼠标移动到链接上 */
a:active {color: orange}           /* 选定的链接 */
</style>
</head>
<body>
<a href = "">链接到本页</a>
<a href = "http://www.sohu.com">搜狐</a>
</body>
</html>
```

在浏览器中的效果如图 4.18 所示。鼠标停留在超级链接上时显示颜色为蓝色；访问过后显示颜色为绿色。

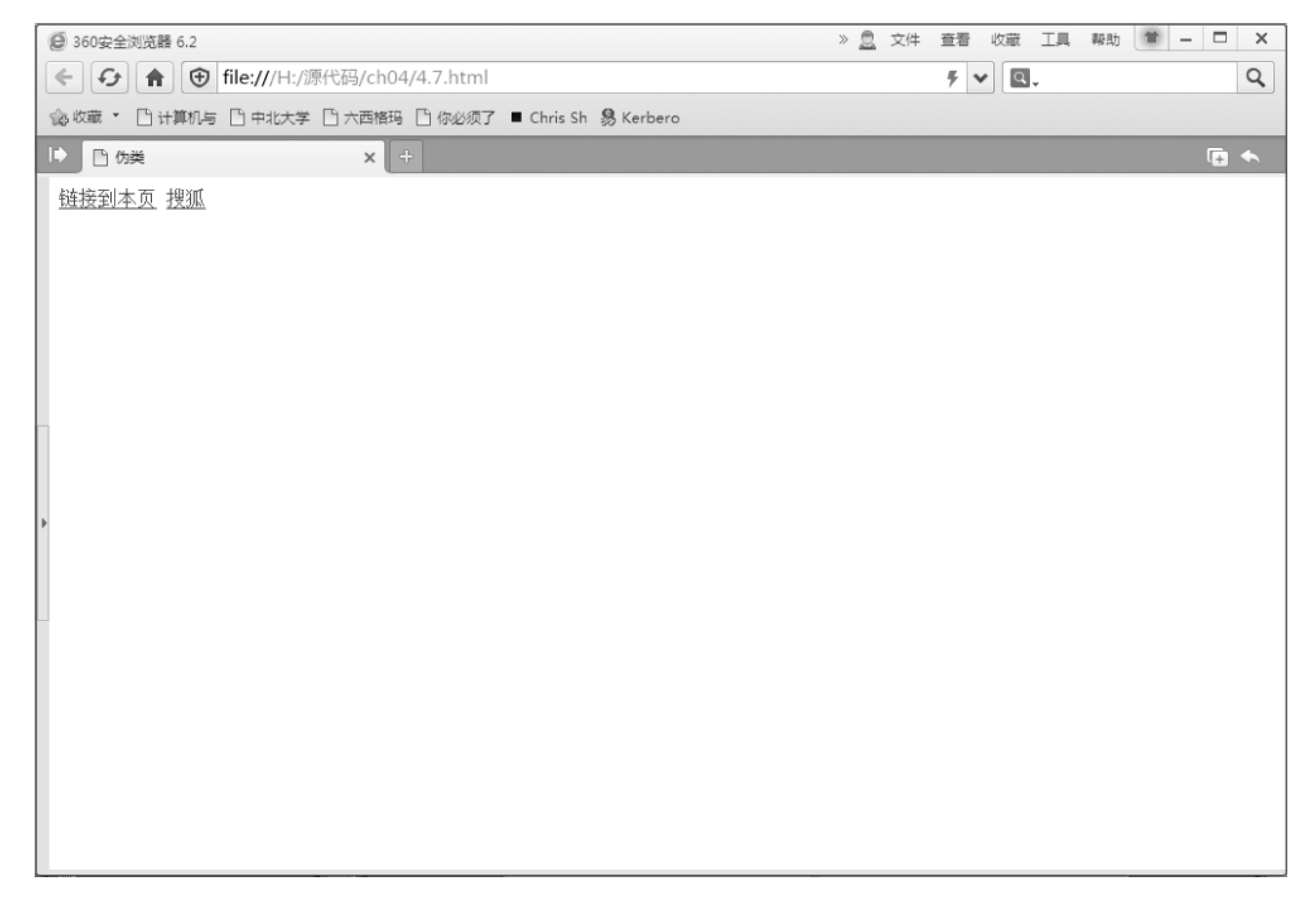


图 4.18 浏览器显示效果

8. 属性选择器

属性选择器即直接使用属性控制 HTML 标记样式。它是根据某个属性是否存在或属性值来寻找元素，因此能够实现某些非常有意思和强大的效果。在 CSS3 版本中，常用的属性选择器如表 4.1 所示。

表 4.1　CSS3 属性选择器

格　式	说　明
E[foo]	选择匹配 E 的元素，且该元素定义了 foo 属性。注意，E 选择器可以省略，表示选择定义了 foo 属性的任意类型元素
E[foo="bar"]	选择匹配 E 的元素，且该元素将 foo 属性值定义为"bar"。注意，E 选择器可以省略，用法与上一个选择器类似
E[foo～="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值是一个以空格符分隔的列表，其中一个列表的值为"bar"。注意，E 选择符可以省略，表示可以匹配任意类型的元素。 例如：a[title～="b1"]匹配，而不匹配
E{foo|="en"}	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值是一个用连字符(-)分隔的列表，值开头的字符为"en"。注意，E 选择符可以省略，表示可以匹配任意类型的元素。例如：[lang|="en"]匹配<body lang="en-us"></body>，而不是匹配<body lang="f-ag"></body>
E[foo ^="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含前缀为"bar"的子字符串。注意，E 选择符可以省略，表示可以匹配任意类型的元素。例如：body[lang ^="en"]匹配<body lang="en-us"></body>，而不匹配<body lang="f-ag"></body>
E[foo $ ="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含后缀为"bar"的子字符串。注意 E 选择符可以省略，表示可以匹配任意类型的元素。例如：img[src $ ="jpg"]匹配，而不匹配
E[foo * ="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 值包含"b"的子字符串。注意 E 选择符可以省略，表示可以匹配任意类型的元素。例如：img[src * ="jpg"]匹配，而不匹配

【例 4.12】 属性选择器实例。

```
<html>
<head>
<title>属性选择器</title>
<style>
[align]{color:red}
[align="left"]{font-size:20px;font-weight:bolder;}
[lang^="en"]{color:blue;text-decoration:underline;}
[src$ ="gif"]{border-width:5px;boder-color:#ff9900}
</style>
</head>
<body>
<p align=center>这是使用属性定义样式</p>
<p align=left>这是使用属性值定义样式</p>
<p lang="en-us">此处使用属性值前缀定义样式</p>
<p>下面使用了属性值后缀定义样式
<img src="2.gif" border="1"/>
</body>
</html>
```

在浏览器中的效果如图 4.19 所示。



图 4.19 浏览器显示效果

9. 结构伪类选择器

结构伪类是 CSS3 新增的类型选择器。即利用文档结构树(DOM)实现元素过滤,通俗地说,就是通过文档结构的相互关系来匹配特定的元素,从而减少文档内对 class 属性和 ID 属性的定义,使得文档更加简洁。

常见结构伪类选择器如表 4.2 所示。

表 4.2 常见结构伪类选择器

选 择 器	含 义
E:root	匹配文档的根元素,对于 HTML 文档,就是 HTML 元素
E:nth-child(n)	匹配其父元素的第 n 个子元素,第一个编号为 1
E:nth-last-child(n)	匹配其父元素的倒数第 n 个子元素,第一个编号为 1
E:nth-of-type(n)	与:nth-child()作用类似,但是仅匹配使用同种标签的元素
E:nth-last-of-type(n)	与:nth-last-child()类似,但是仅匹配使用同种标签的元素
E:last-child	匹配父元素的最后一个子元素,等同于:nth-last-child(1)
E:first-of-type	匹配父元素下使用同种标签的第一个子元素,等同于:nth-of-type(1)
E:last-of-type	匹配父元素下使用同种标签的最后一个子元素,等同于:nth-last-of-type(1)
E:only-child	匹配父元素下仅有的一个子元素,等同于:first-child:last-child 或 nth-child(1):nth-last-child(1)
E:only-of-type	匹配父元素下使用同种标签的唯一一个子元素,等同于:first-of-type:last-of-type 或:nth-of-type(1):nth-last-of-type(1)
E:empty	匹配一个不包含任何子元素的元素,注意文本结点也被看作子元素

【例 4.13】 结构伪类选择器实例。

```
<html>
<head>
<title>结构伪类</title>
<style>
tr:nth-child(even){
background-color:#f5fafe
}
tr:last-child{font-size:20px;}
</style>
</head>
<body>
<table border=1 width=80%>
<th>姓名</th><th>编号</th><th>性别</th>
<tr><td>刘海松</td><td>006</td><td>男</td></tr>
<tr><td>王峰</td><td>001</td><td>女</td></tr>
<tr><td>李张力</td><td>002</td><td>男</td></tr>
<tr><td>于辉</td><td>008</td><td>男</td></tr>
<tr><td>张浩</td><td>004</td><td>女</td></tr>
<tr><td>刘永权</td><td>003</td><td>男</td></tr>
</table>
</body>
</html>
```

浏览器的效果如图 4.20 所示。

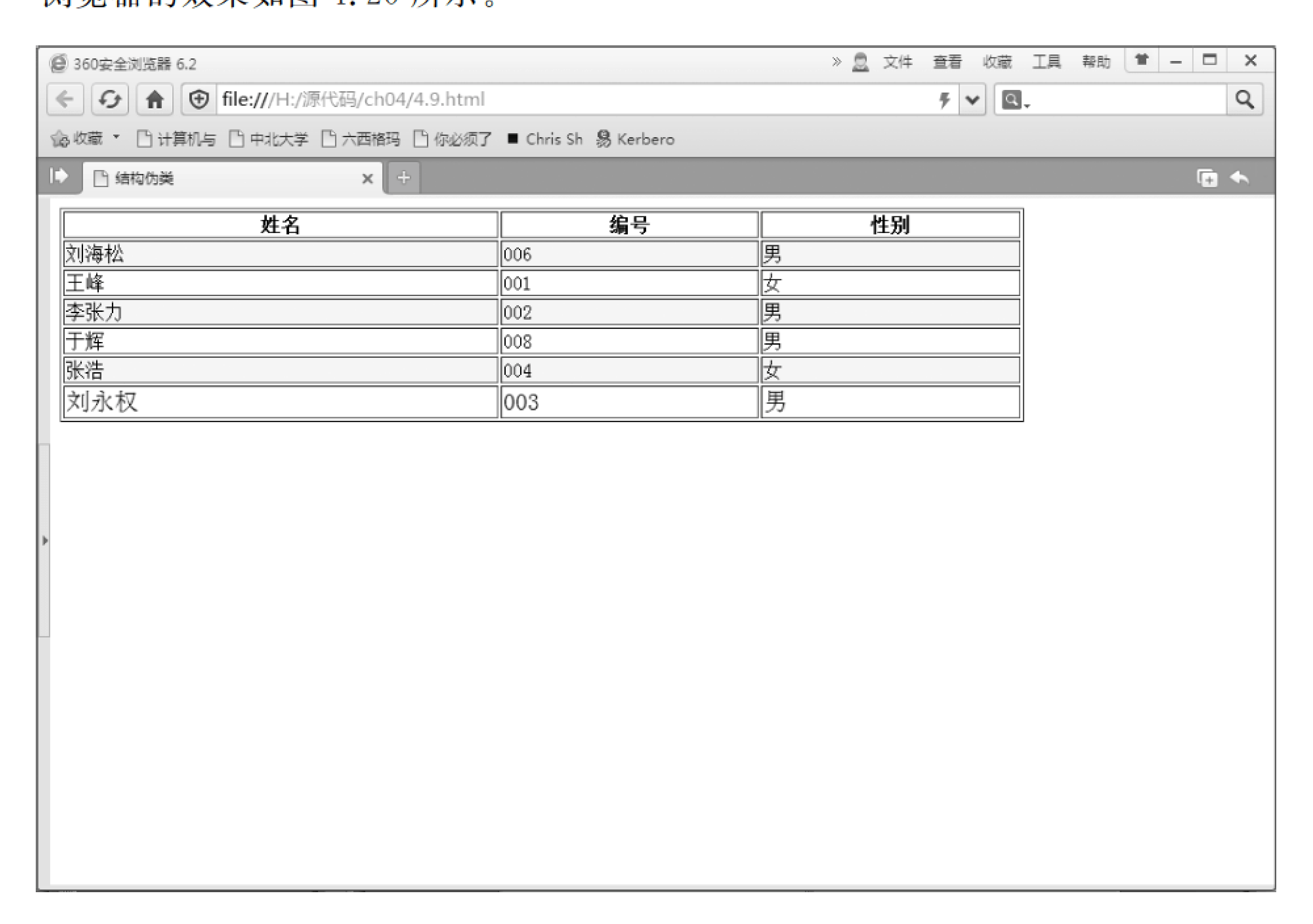


图 4.20　浏览器显示效果

10. UI 元素状态伪类选择器

UI 元素状态伪类也是 CSS3 新增选择器。其中，UI 即 User Interface(用户界面)的缩写。UI 设计则是对软件的人机交互、操作逻辑、界面美观的整体设计。好的 UI 设计不仅要使软件变得有个性品味，还要让软件的操作变得舒适、简单、自由，充分体现软件的定位和特点。

UI 元素的状态一般包括可用、不可用、选中、未选中、获得焦点、失去焦点、锁定、待机等。CSS3 定义了三种常见的状态伪类选择器，详细说明如表 4.3 所示。

表 4.3 UI 元素状态伪类表

选择器	说明
E:enabled	选择匹配 E 的所有可用 UI 元素。注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如：input:enabled 匹配＜form＞＜input type＝text/＞＜input type＝button disabled＝"disabled"/＞＜/form＞代码中的文本框，而不匹配代码中的按钮
E:disabled	选择匹配 E 的所有不可用元素。注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如：input:disabled 匹配＜form＞＜input type＝text/＞＜input type＝button disabled＝"disabled"/＞＜/form＞代码中的按钮，而不匹配代码中的按钮
E:checked	选择匹配 E 的所有可用 UI 元素。注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如：input:checked 匹配＜form＞＜input type＝checkbox/＞＜input type＝radio checked＝"checked"/＞＜/form＞代码中的单选按钮，而不匹配代码中的按钮

【例 4.14】 UI 元素状态伪类选择器实例。

```
<html>
<head>
<title>UI 元素状态伪类选择器</title>
<style>
input:enabled {    border:1px dotted #666;    background:#ff9900;    }
input:disabled {    border:1px dotted #999;    background:#F2F2F2;    }
</style>
</head>
<body>
<center>
<h3 align = center>用户登录</h3>
<form method = "post" action = "">
用户名: <input type = text name = name><br>
密   码: <input type = password name = pass disabled = "disabled"><br>
<input type = submit value = 提交>
<input type = reset value = 重置>
</form>
<center>
</body>
</html>
```

在浏览器中的显示效果如图 4.21 所示。



图 4.21 浏览器显示效果

4.2.2 选择器声明

1. 集体声明

在一个页面中,有时不同的标记样式要保持一致。例如,p 标记和 h1 标记要保持一致,此时可以将 p 标记和 h1 标记共同使用类选择器,除了这个方法外,还可以使用集体声明方法。集体声明就是在声明各种 CSS 选择器时,如果某些选择器的风格是完全相同或者部分相同的,可以将风格相同的 CSS 选择器同时声明。

【例 4.15】 集体声明实例。

```
<html>
<head>
<title>集体声明</title>
<style type = "text/css">
h1,h2,p{
color:red;
font - size:20px;
font - weight:bolder;
}
</style>
</head>
<body>
<h1>此处使用集体声明</h1>
<h2>此处使用集体声明</h2>
<p>此处使用集体声明</p>
</body>
```

```
</html>
```

在浏览器中的效果如图 4.22 所示。

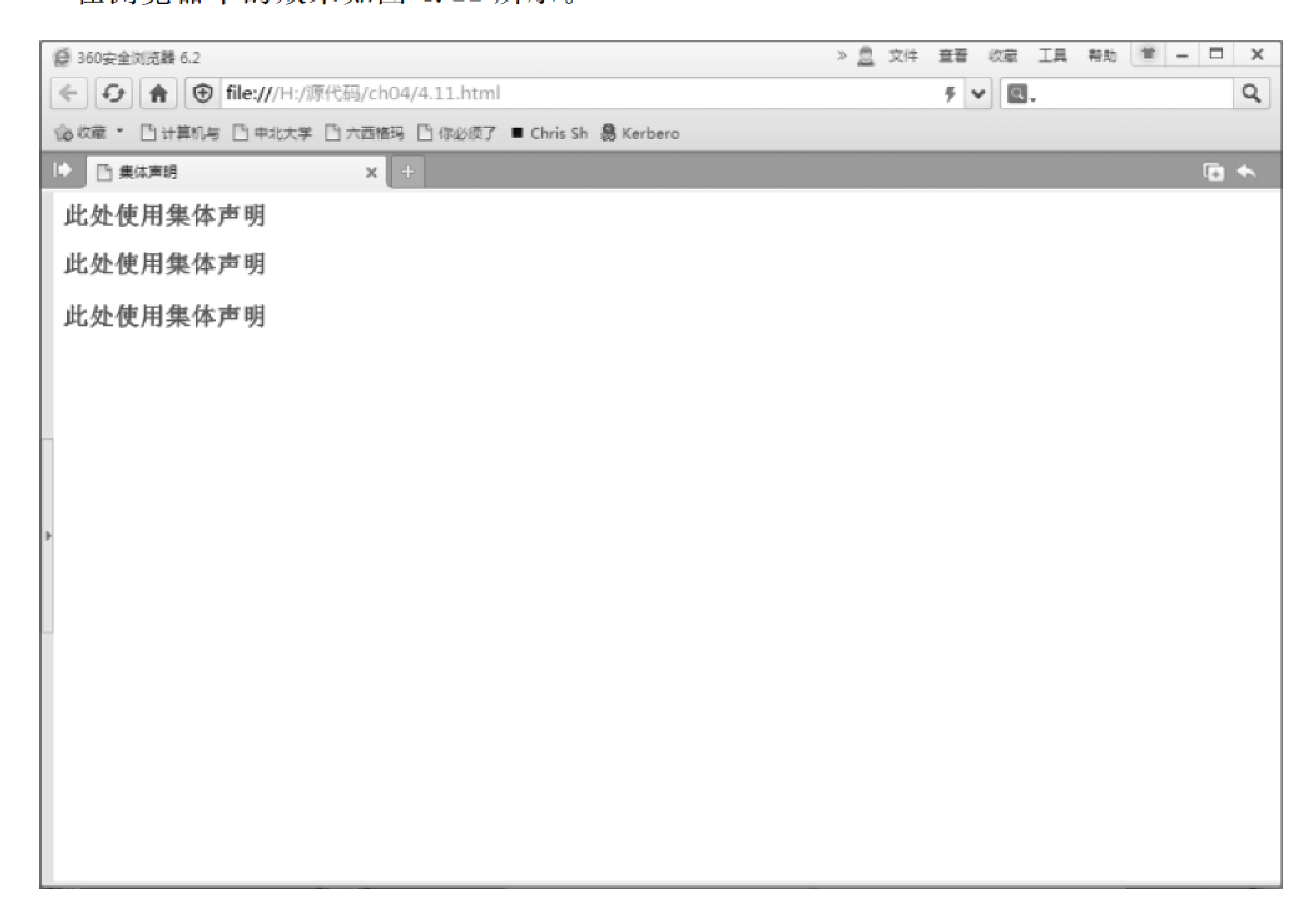


图 4.22 浏览器显示效果

2. 多重嵌套声明

在 CSS 控制 HTML 标记样式时，还可以使用层层递进的方式，即嵌套方式，对指定位置的 HTML 标记进行修饰。例如，当<p>与</p>之间包含<a>标记时，就可以使用这种方式对 HTML 标记进行修饰。

【例 4.16】 多重嵌套声明实例。

```
<html>
<head>
<title>多重嵌套声明</title>
<style>
p{font-size:20px;}
p a{color:red;font-size:30px;font-weight:bolder;}
</style>
</head>
<body>
<p>这是一个多重嵌套<a href="">测试</a></p>
</body>
</html>
```

在浏览器中的效果如图 4.23 所示。

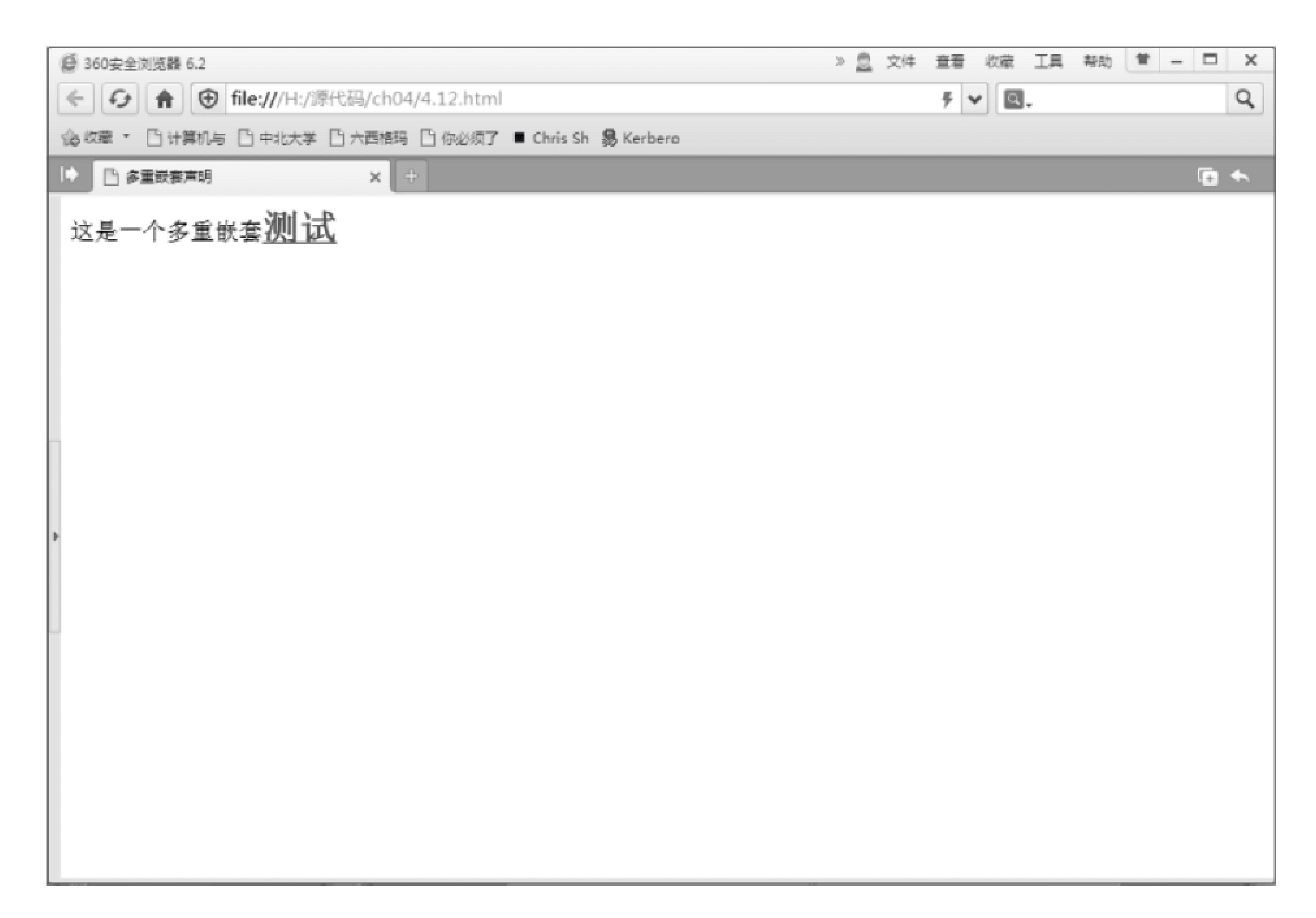


图 4.23　浏览器显示效果

4.2.3　CSS3 常用单位

1. 颜色单位

通常颜色用于设定字体以及显示背景色。在 CSS3 中颜色设置的方法很多，有以下几种方式。

1）命名颜色

CSS3 可以直接用英文单词命名与之相应的颜色，这种方法最简单、直接、容易掌握。如 red、yellow、black、green 等。例如：

```
P{color:red}
```

2）RGB 色彩模式

使用 RGB 颜色的格式必须使用 rgb(R、G、B)，其中，R、G、B 分别表示红、绿、蓝的十进制值，其取值范围为 0～255，通过这三个值的变化组合便可形成不同的颜色。

RGB 设置方式一般分为两种：百分比设置和直接用数值设置。例如：

```
p{color:rgb(123,0,25)}
p{color:rgb(45%,0%,25%)}
```

3）十六进制颜色

十六进制颜色是最常用的定义方式。其基本格式为＃RRGGBB，其中 R 表示红色，G 表示绿色，B 表示蓝色。而 RR、GG、BB 最大值为 FF，表示十进制中的 255；最小值为 00，

表示十进制中的 0。例如：

```
p{color:ffff00}
```

4) HSL 色彩模式

HSL 色彩模式是工业界的一种颜色标准，它通过对色调(H)、饱和度(S)、亮度(L)三个颜色通道的改变以及它们相互之间的叠加来获得各种颜色。

在 CSS3 种，HSL 色彩模式的表示语法如下：

```
hsl(<length>,<percentage>,<percentage>)
```

其中，length 表示色调，可以取任意数值，取值范围为 0～360。第一个 percentage 表示饱和度，取值范围为 0～100%。第二个 percentage 表示亮度，取值范围为 0～100%。

其使用示例如下：

```
p{color:hsl(10,80%,80%);}
```

5) HSLA 色彩模式

HSLA 是 HSL 色彩模式的扩展，在色相、饱和度、亮度三要素的基础上增加了不透明度参数。其语法格式如下：

```
hsla(<length>,<percentage>,<percentage>,<opacity>)
```

其中，前三个参数与 hsl()函数的参数的意义和用法相同，第 4 个参数<opacity>表示不透明度，取值在 0～1 之间。例如：

```
p{color:hsla(10,80%,80%,0.9);}
```

6) RGBA 色彩模式

RGBA 是 RGB 色彩模式的扩展，在红、绿、蓝三原色的基础上增加了不透明度参数。其语法格式为：

```
rgba(r,g,b,<opacity>)
```

使用示例如下：

```
p{color:rgba(23,128,52,0.9);}
```

7) 网络安全色

网络安全色由 216 种颜色组合而成，被认为在任何操作系统和浏览器中都是相对稳定的，也就是说显示的颜色是相同的，因此，这 216 种颜色被称为网络安全色。这 216 种颜色都是由红绿蓝三种基本色从 0、51、102、153、204、255 这 6 个数值中取值，组成 6×6×6 种颜色。

2. 长度单位

CSS 中，长度单位可以被分为两类：绝对单位和相对单位。

1) 绝对单位

(1) 英寸(in)：英寸对于国内而言，使用较少，是国外常用的度量单位。1in＝2.54cm，

1cm＝0.394in。

(2) 厘米(cm)：可以用来设定距离比较大的页面元素框。

(3) 毫米(mm)：可以用来比较精确地设定页面元素距离或大小。10mm＝1cm。

(4) 磅(pt)：一般用来设定文字的大小。它是标准的印刷量度，广泛应用于打印机、文字程序等。72pt＝1in＝2.54cm。

(5) pica(pc)：是另一种印刷量度。1pica＝12pt。该单位不常用。

2) 相对单位

相对单位是指在量度时需要参照其他页面元素的单位值。使用相对单位所量度的实际距离可能会随着这些单位值的改变而改变。CSS 提供了三种相对单位：em、ex 和 px。

(1) em：用于给定字体的 font-size 值。例如，一个元素字体大小为 12pt，那么 1em 就是 12pt；如果该元素字体大小改为 15pt，则 1em 就是 15pt。简单来说，1em 总是字体的大小值。em 的值总是随着字体大小的变化而变化的。

例如，分别设定页面元素 h1、h2 和 p 的字体大小为 20pt、15pt 和 10pt，各元素的左边距为 1em，样式规则如下：

```
h1{font - size:20pt}
h2{font - size:15pt}
p{font - size:10pt}
h1,h2,p{margin - left:1em}
```

对于 h1，1em 等于 20pt；对于 h2，1em 等于 15pt；对于 p，1em 等于 10pt，所以 em 的值会随着相应元素字体大小的变化而变化。另外，em 值有时还相对于其上级元素的字体大小。例如，上级元素字体大小为 20pt，设定其子元素字体大小为 0.5em，则子元素显示出的字体大小则为 10pt。

(2) ex：ex 是以给定字体的小写字母“x”高度作为基准的，对于不同字体来说，小写字母“x”高度是不同的，所以 ex 单位的基准也不同。

(3) px：px 也叫像素，这是目前使用最为广泛的一种单位，1px 也就是屏幕上的一个小方格。由于显示器有多种不同的大小，它的每个小方格大小也是不同的，所以像素单位的标准也都不一样。

4.2.4　CSS 继承

CSS 继承指的是子标记会继承父标记的所有样式风格，并可以在父标记样式风格的基础上加以修改，产生新的样式，而子标记样式风格完全不会影响父标记。

【例 4.17】　多重嵌套声明实例。

```
<html>
<head>
<title>多重嵌套声明</title>
<style>
p{font - size:20px;color:red}
span{font - size:30px;}
</style>
</head>
```

```
<body>
<p>这是一个继承<span>测试</span></p>
</body>
</html>
```

浏览器中的效果如图 4.24 所示。



图 4.24　浏览器显示效果

4.3　在 Dreamweaver 中使用 CSS 样式

在 Dreamweaver 中，利用 CSS 样式表可以设置丰富的样式，包括文字样式、背景样式和边框样式等各种常用效果，这些样式决定了页面中的文字、列表、背景、表单、图片和光标等各种元素。

4.3.1　Dreamweaver 中的样式类型

在 Dreamweaver 中，CSS 主要通过 CSS 样式面板实现。创建新的 CSS 规则的步骤如下。

(1) 选择“窗口”|“CSS 样式”菜单命令，打开“CSS 样式”面板，如图 4.25 所示。

(2) 在“CSS 样式”面板中，单击右下角的“新建 CSS 规则”按钮，打开“新建 CSS 规则”对话框，如图 4.26 所示。

(3) 在该对话框中，可以定义创建的 CSS 样式的类型。

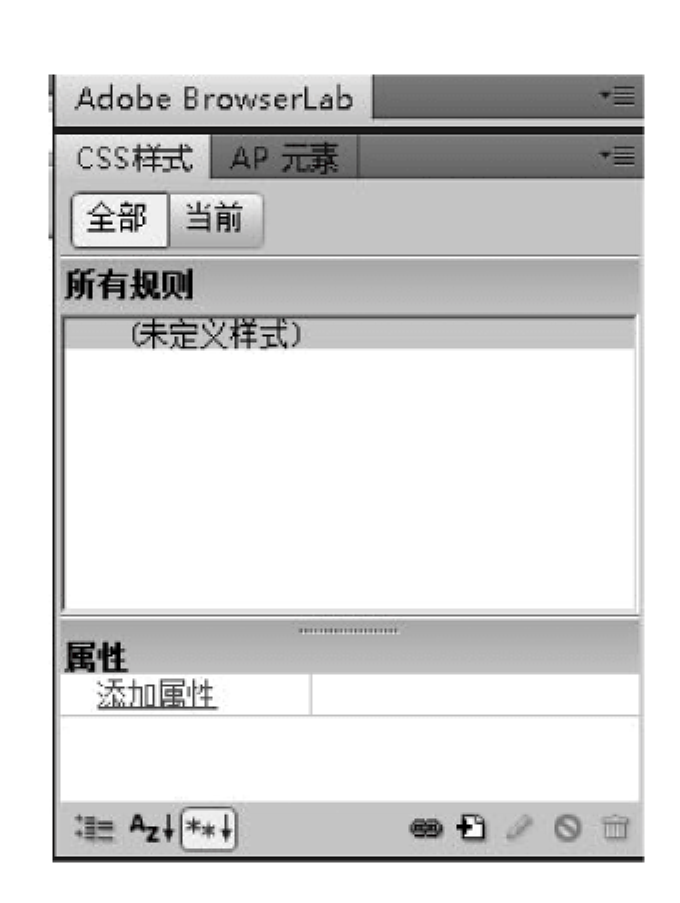




图 4.25 “CSS 样式”面板



图 4.26 “新建 CSS 规则”对话框

选择器类型有以下几种。

① “类(可应用于任何 HTML 元素)”：创建一个可作为 class 属性应用于任何 HTML 元素的自定义样式。

② “ID(仅用于一个 HTML 元素)”：定义包含特定 ID 属性的标签的格式。可以从“选择器类型”弹出菜单中选择 ID 选项，然后在“选择器名称”文本框中输入唯一 ID。

③ “标签(重新定义 HTML 元素)”：重新定义特定 HTML 标签的默认格式设置，可选择重新定义标签。

④ “符合内容(基于选择的内容)”：定义同时影响两个或多个标签、类或 ID 的复合规则。例如，输入 DIV p，则 DIV 标签内的 p 元素都将受此规则的影响。说明文本区域准确说明用户添加或删除选择器时该规则将影响哪些元素。

选择器名称：类名称必须以句点开头，可以包含任何字母和数字的组合(如 .mystyle)。如果没有输入开头的句点，Dreamweaver 会自动添加；ID 名称必须以“#”开头，可以包含任何字母和数字的组合(如 #myID)，如果没有输入开头的“#”，Dreamweaver 会自动添加。

规则定义：选中“仅限该文档”选项，则定义的 CSS 规则只对当前文档起作用，不保存编辑的样式；选中“新建样式表文件”选项，可以定义一个外部链接的 CSS 样式。

4.3.2 CSS 类型设置

在“新建 CSS 规则”对话框中设置所需的样式后，单击“确定”按钮，打开 CSS 规则定义对话框。在该对话框中，可以设置样式属性，其中定义样式最常使用的属性包含在“类型”选项面板中，如图 4.27 所示。

(1) Font-family(字体)：在下拉列表中可以选择当前样式所用的字体，设置以逗号分开的字体名称列表。当浏览器解析执行时，会将所列字体按顺序从前到后选择，即当客户机中没有第一个字体时，浏览器自动利用第二种字体，以此类推。

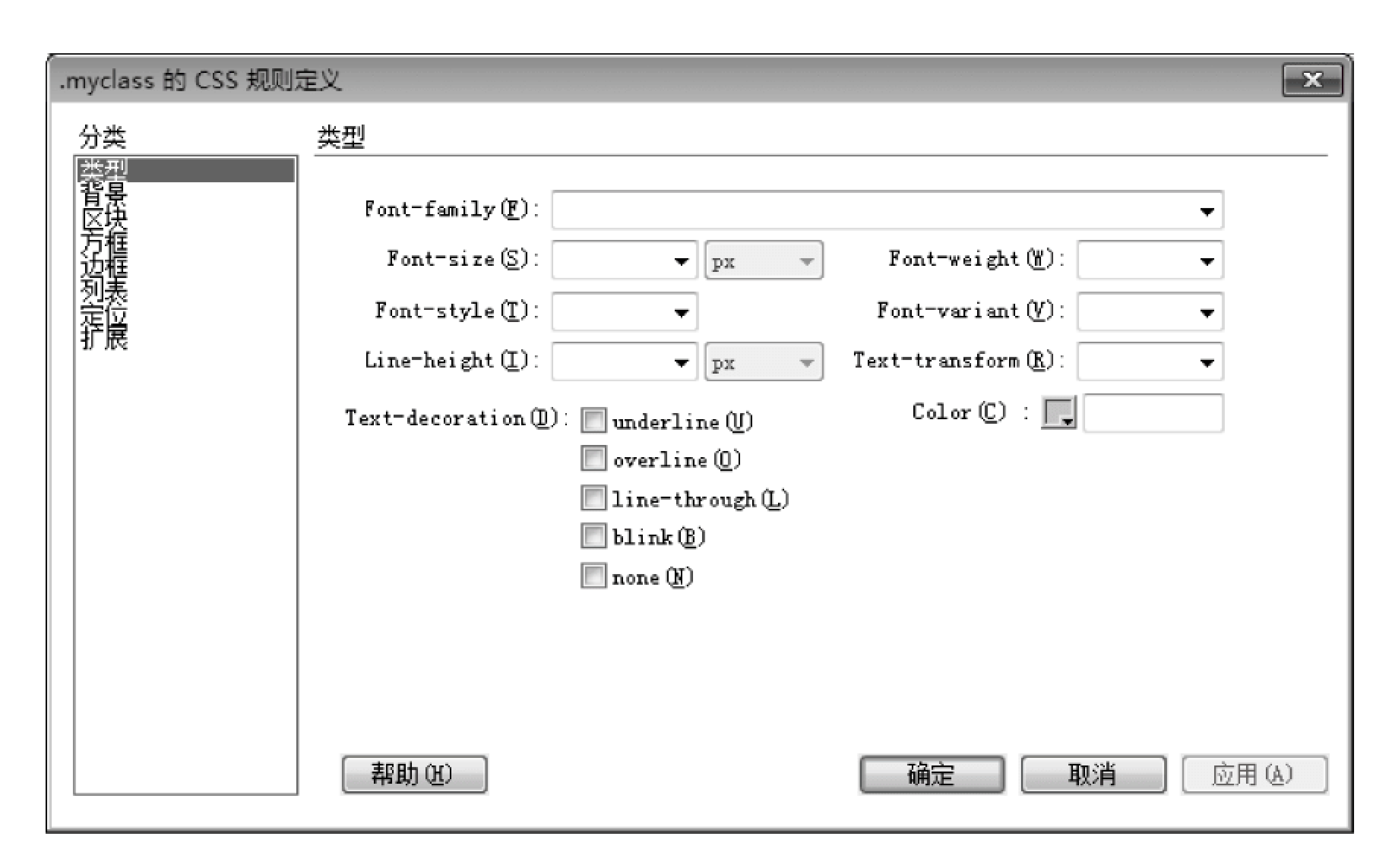


图 4.27 “类型”选项面板

(2) Font-size(字号):在列表中设置字体的字号。可以选择或输入具体的数值,也可以指定绝对大小的字号;如果选择大或小之类的选项,则设置的是字体的相对大小。如果要设置字体的绝对大小,则需要在其右侧的下拉列表中选择单位。

(3) Font-style(文字样式):在下拉列表中选择字体的格式。可以选择 normal(正常)、italic(斜体)或 oblique(偏斜体)选项

(4) Line-height(行高):通常情况下,浏览器会用单行距离。在该下拉列表中,可以选择文本的行高,若选择 normal(正常)选项,则由系统自动计算行高和字体大小;如果希望具体指定行高值,在其中输入需要的数值,然后选择单位即可。

(5) Text-decoration(文字修饰):在该区域可以设置字体的修饰格式,包括 underline(下划线)、overline(上划线)、line-through(删除线)和 blink(闪烁线)等。在默认状态下,普通文本的修饰格式为 none(无),超级链接的修饰格式为 underline(下划线)。

(6) Font-weight(字体粗细):在该下拉列表中指定字符的粗细。选择或输入数值,可以指定字体的绝对粗细程度;使用 bolder 和 lighter 值可得到比父元素字体更粗或更细的字体。

(7) Font-variant(字体变形):在该下拉列表中设置字体的变体形式。可以选择所需字体的某种变形。这个属性的默认值是 normal,表示字体的常规版本。也可以指定 small-caps 来选择字体的形式,在这个形式中,小写字母都会被替换为小的大写字母。但在文档窗口中不能直接看到设置结果,必须在浏览器中才能看到效果。

(8) Text-transform(文字大小写):在该下拉列表中设置字符的大小写方式。如果选择 capticalize(首字母大写)选项,则可以指定将每个单词的第一个字母大写;如果选择 uppercase(大写)或 lowercase(小写)选项,则可以分别将所有被选择的文本都设置为大写或小写;如果选择 none(无)选项,则保持字符本身原有的大小写格式。

(9) Color(颜色):单击该按钮,可以打开色板,设置 CSS 样式的字符颜色。

4.3.3　CSS 背景设置

利用样式表不仅可以设置整个文档的背景，还可以设置特定部分的背景色或背景图像。插入背景图像时，可以指定背景图像的位置或重复方式、固定与否等。在“新建 CSS 规则”对话框中设置所需的样式后，单击“确定”按钮，打开 CSS 规则定义对话框。在该对话框的“背景”选项面板中可以定义背景样式，如图 4.28 所示。

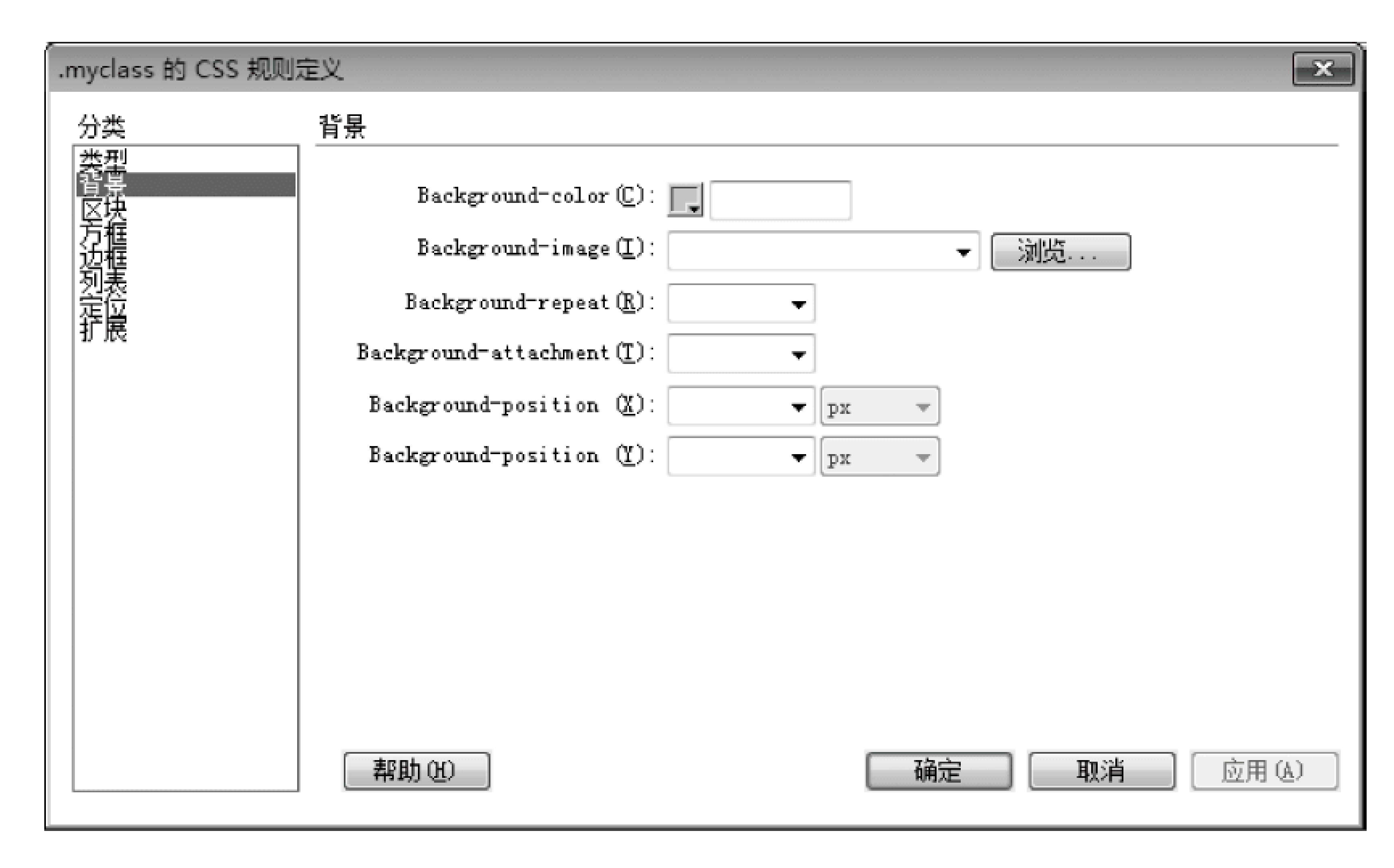


图 4.28　“背景”选项面板

(1) Background-color(背景颜色)：用来设定页面背景色。

(2) Background-image(背景图像)：用来设定页面背景图像。单击“浏览”按钮可以在对话框中选择图像源文件。

(3) Background-repeat(背景重复)：用于设定使用图像当背景时重复显示的形式，一般用于图片面积小于页面元素面积的时候。共有 4 个选项：no-repeat(不重复)表示只在应用样式的元素前端显示一次该图像；repeat(重复)表示在应用样式的元素背景上的水平方向和垂直方向上均重复显示该图像；repeat-x(横向重复)表示在应用样式的元素背景上的水平方向上重复显示该图像；repeat-y(纵向重复)表示在应用样式的元素背景上的垂直方向上重复显示该图像。

(4) Background-attachment(背景固定)：其中的 fixed(固定)与 scroll(滚动)选项用来设定对象的背景图像是随对象内容滚动的还是固定的。

(5) Background-position(X)(水平位置)/ Background-position(Y)(垂直位置)：用于指定背景图像相对于应用样式的元素的水平位置或垂直位置。可以选择 left(左对齐)、right(右对齐)、center(居中对齐)或 top(顶部对齐)、bottom(底部对齐)选项，也可以直接输入数值。如果前面的附件选项设置为 fixed，则元素的位置是相对于文档窗口，而不是元素本身的。可以为 Background-position 属性指定一个或两个值，如果用的是一个值，它将同时应用于水平和垂直位置；如果是两个值，那么第一个值表示水平偏移，第二个值表示垂直偏移。

默认情况下,能够识别样式的浏览器将从分配的显示区域的左上角开始放置背景图像,并将图像平铺至同一区域的右下角。

4.3.4 CSS 区块设置

选择规则定义对话框左侧的“区块”选项,在右侧选项面板中可以对字间距及排列方式进行设置,如图 4.29 所示。

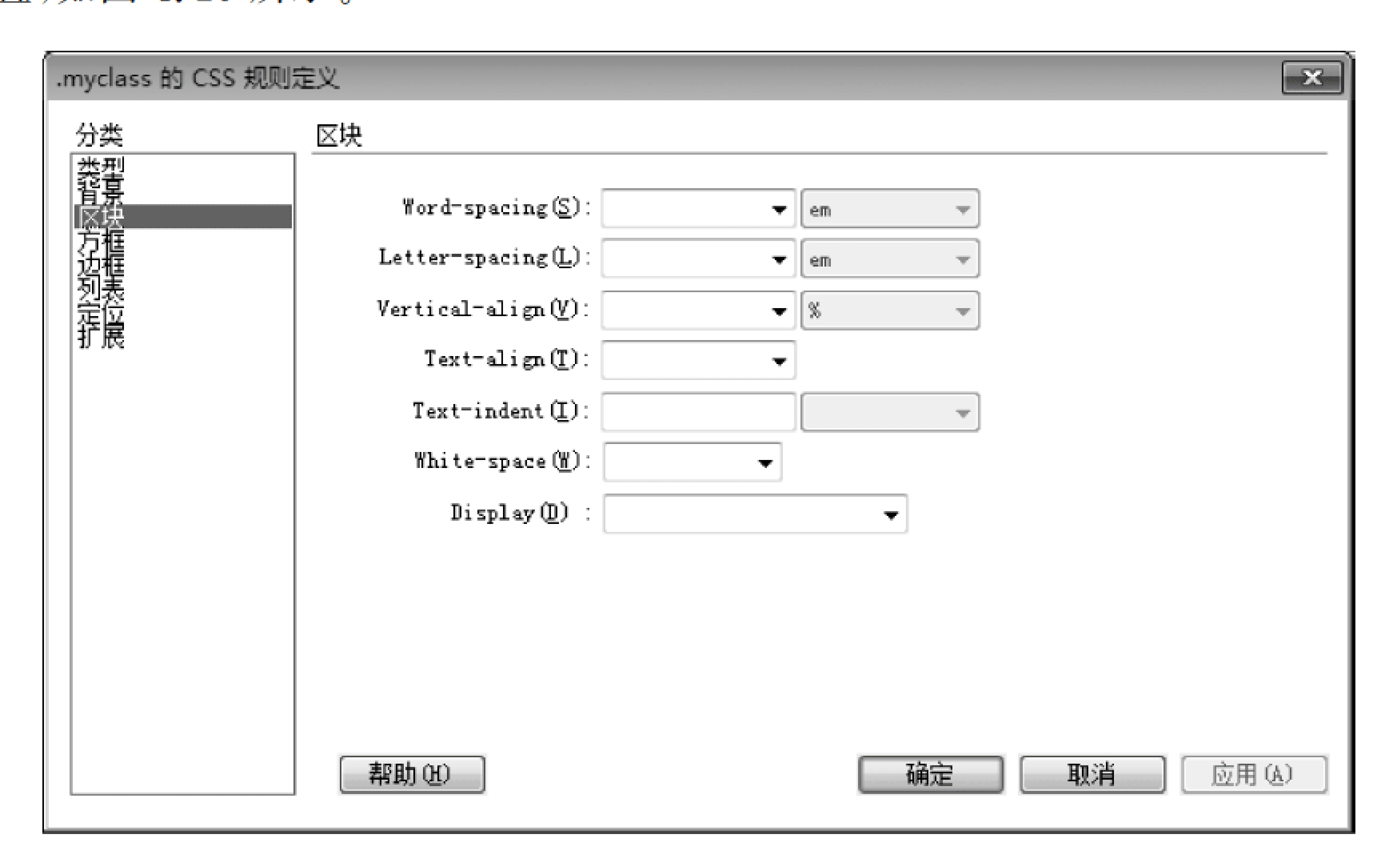


图 4.29 “区块”选项面板

(1) Word-spacing(单词间距):在字与字之间增加更多的空隙,可以在单词之间添加额外的间距。可以输入一个值,然后在右侧的下拉列表中选择数值的单位,但这要根据浏览器决定,很多浏览器并不支持负值。

(2) Letter-spacing(字母间距):调整字符之间的间距。与单词间距的设置相同,此选项也可以在字符之间添加额外的间距。用户可以输入一个值,然后在右侧的下拉列表中选择数值的单位,但这要根据浏览器决定,很多浏览器并不支持负值。另外,字母间距的优先级高于单词间距。

(3) Vertical-align(垂直对齐):调整页面元素的垂直位置,多数情况下要参照其父对象的位置。

(4) Text-align(文本对齐):文本相对页边的调整几乎是所有文字处理器都具有的基本特性。Text-align 属性使得 HTML 的任何模块级标签都具有了这个能力。定义对象的对齐方式是 left(居左)、right(居右)、center(居中)或 justify(绝对居中)。

(5) Text-indent(文字缩进):设置每段第一行的缩进距离,允许负值,但有些浏览器不支持。

(6) White-space(空格):决定一个元素怎样处理其中的空白部分,有三个属性值。mormal(正常)是按照正常的方法处理空格,可以使多重的空白合并成一个;pre(保留)是保留应用样式元素中空格的原始形象,不允许多重的空白合并成一个;nowrap(不换行)是长文本不自动换行。

(7) Display(显示)：指定是否及如何显示元素。选择 none 选项将关闭该样式被指定给的元素的显示。

4.3.5　CSS 方框设置

文档中的每个元素都可以装在一个矩形框内，通过 CSS 可以控制框的大小、位置和外观。在 CSS 规则定义对话框的“方框”选项面板中可以定义方框的样式，如图 4.30 所示。



图 4.30　“方框”选项面板

(1) Width(宽)：在该下拉列表中可以选择元素的宽度。auto(自动)选项表示由浏览器自行控制，也可以直接输入一个值，并在右侧的下拉列表中选择值的单位。只有当该样式应用到图像或分层上面时，才可以直接从文档窗口中看到设置结果。

(2) Height(高)：在该下拉列表中可以选择元素的高度。auto(自动)选项表示由浏览器自行控制，也可以直接输入一个值，并在右侧的下拉列表中选择值的单位。只有当该样式应用到图像或分层上面时，才可以直接从文档窗口中看到设置结果。

(3) Float(浮动)：在该下拉列表中可以选择应用样式的元素的浮动位置。利用该选项，可以将元素移动到页面范围之外，如果选择 left 选项，则将元素放置到左页面空白处；如果选择 right 选项，则将元素放置到右页面空白处。

(4) Clear(清除)：在该下拉列表中可以定义不允许分层。如果选择 left 选项，则表明不允许分层出现在应用该样式元素的左侧；如果选择 right 选项，则表明不允许分层出现在应用该样式元素的右侧。如果分层出现在元素相应的那侧，则该元素会在分层下自动移开。

(5) Padding(填充)：在该区域可以定义应用样式的元素内容和元素边界之间的空白大小。可以分别在上、下、左、右 4 个下拉列表中输入相应的值，然后在右侧的下拉列表中选择适当的数值单位。padding-left、padding-right、padding-top、padding-bottom 属性都接受长度或百分比值，指定元素周围要保留多少空白。

(6) Margin(边界)：在该区域可以定义应用样式的元素边界和其他元素之间的空白大小。同样可以分别在上下左右 4 个下拉列表中输入相应的值，然后在右侧的下拉列表中选

择适当的数值单位。margin-left、margin-right、margin-top、margin-bottom 属性都接受长度或百分比值,指定元素周围要保留多少空白。

这种方式控制对象来排版的方法就是"DIV+CSS 结构",它是 Web 标准所推荐的排版方式,目前大多数的网站都是采用这种方式来进行页面布局的。

4.3.6 CSS 边框设置

边框属性是一个用来设置元素边框的宽度、式样和颜色的缩写。在 CSS 规则定义对话框的"边框"分类中定义边框样式,如图 4.31 所示。



图 4.31 "边框"属性面板

(1) Style(样式):有 9 个选项,每个选项代表一种边框样式。Border-style 属性值包括 none(默认)、dotted、dashed、solid、double、groove、ridge、inset 和 outset。

(2) Width(宽度):可以定义应用该样式的元素的边框宽度。在 Top、Right、Bottom 和 Left 4 个下拉列表中,可以分别设置边框上每个边的宽度。用户可以选择相应的宽度选项,如细、中、粗或具体数值来指定。其他方向的边框宽度设定都与"上"相同。

(3) Color(颜色):可分别设定上下左右边框的颜色,或勾选"全部相同"复选框为所有边线设置相同颜色。

4.3.7 CSS 列表设置

选择规则定义对话框中左侧的"列表"选项,在右侧选项面板中可以对列表进行设置,如图 4.32 所示。

(1) List-style-type(列表类型):该选项决定了有序和无序列表项如何显示在能识别样式的浏览器上。可为每行的前面加上项目符号或编号,用于区分不同的文本行。

(2) List-style-image(项目符号图像):可以设置以图片作为无序列表的项目符号。可以在其中输入图片文件的 URL 地址,也可以通过单击"浏览"按钮,从磁盘上选择图片

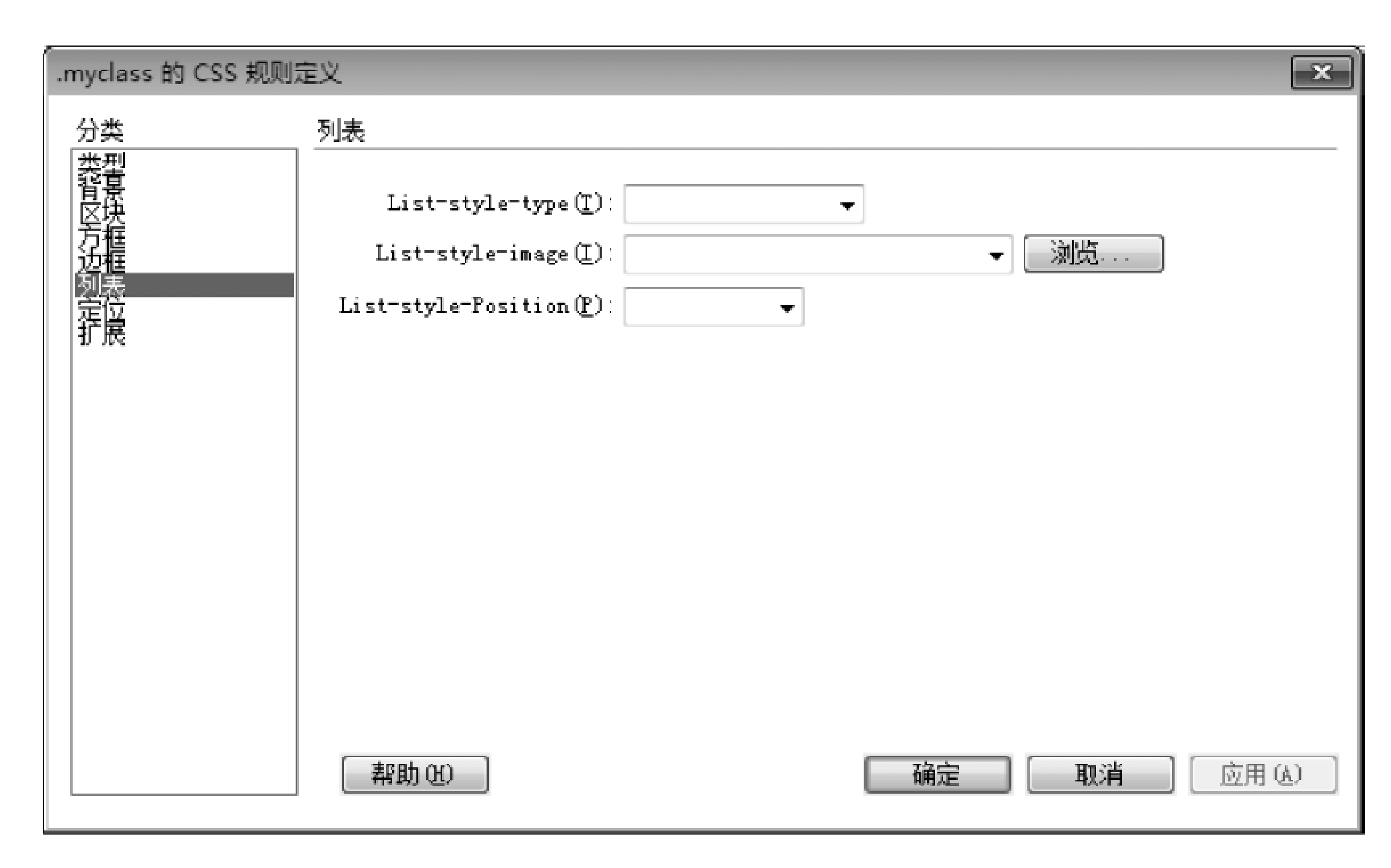


图 4.32　“列表”选项面板

文件。

(3) List-style-Position(位置)：设置列表项的换行位置。有两种方法可用来定位与一个列表项有关的记号，即在与项目有关的块外面或里面。List-style-Position 属性接受以下两个值：inside 或 outside。

4.3.8　CSS 定位设置

在 CSS 规则定义对话框的“定位”选项面板中可以定义定位样式，如图 4.33 所示。

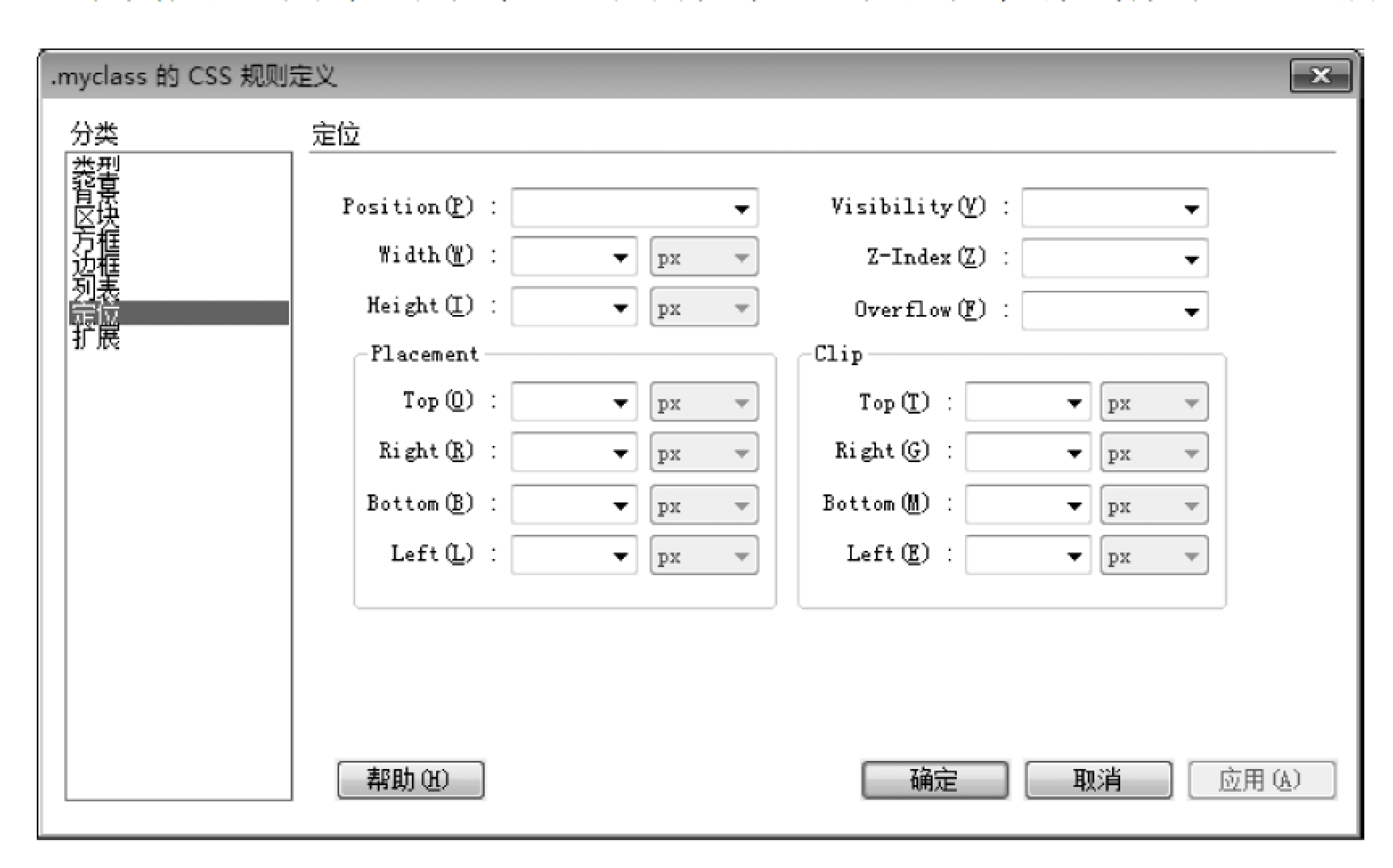


图 4.33　“定位”选项面板

(1) Positon(位置)：可以设置浏览器放置 APDiv 的方式，包含以下 4 项参数。

① Static：应用常规的 HTML 布局和定位规则，并由浏览器决定元素的框的左边缘和

上边缘。

② Relative：使元素相对于其包含的流移动，可以在这种情况下使 Top、Bottom、Left 和 Right 属性都用来计算框相对于其在流中正常位置所处的位置。随后的元素都不会受到这种位置改变的影响，而且放在流中的方式就像没有移动过该元素一样。

③ Absolute：可以从包含文本流中去除元素，而且随后的元素可以相应地向前移动，然后使用 Top、Bottom、Left 和 Right 属性，相对于包含块计算出元素的位置。这种定位允许将元素放在关于其包含元素的固定位置，但会随着包含元素的移动而移动。

④ Fixed：将元素相对于其显示的页面或窗口进行定位。像 Absolute 定位一样，从包含流中去除元素时，其他的元素也会相应发生移动。

(2) Visibility(显示)：设置层的初始化显示位置，包含以下三个选项。inherit(继承)：继承分层父级元素的可视性属性。visible(可见)：不管分层的父级元素是否可见，都显示层内容。hidden(隐藏)：不管分层的父级元素是否可见，都隐藏该层内容。

(3) Width(宽)/Height(高)：设置元素本身的大小。

(4) Z-Index(Z 轴)：定义层的顺序，即层重叠的顺序。可以选择 auto，或输入相应的层索引值。索引值可以为正数或负数，较高值所在的层会位于较低值所在的层的上端。

(5) Overflow(溢出)：定义层中的内容超出了层的边界后发生的情况。包含以下选项：visible(可见)，当层中的内容超出层范围时，层会自动向下或向右扩展大小，以容纳分层内容使之可见；hidden(隐藏)，当层中的内容超出层范围时，层的大小不变，也不出现滚动条，超出层的内容不显示；scroll(滚动)，无论层中的内容是否超出层范围，层上总会出现滚动条，这样即使层内容超出层范围，也可以利用滚动条浏览；auto(自动)，当层中内容超出层范围时，层的大小不变，但会出现滚动条，以便通过滚动条的滚动来显示所有层中内容。

(6) Placement(放置)：设置层的位置和大小。具体的含义基于前面“类型”部分的设置。在 Top、Right、Bottom 和 Left 4 个下拉列表中，可以分别输入相应的值，在右侧的下拉列表中选择相应的数值单位，默认的单位是像素。

(7) Clip(剪辑区域)：定义可视层的局部区域的位置和大小。如果指定了层的碎片区域，则可以通过脚本语言(如 JavaScript)进行操作。在 Top、Right、Bottom 和 Left 4 个下拉列表中，可以分别输入相应的值，在右侧的下拉列表中选择相应的数值单位，默认的单位是像素。

4.3.9 CSS 扩展设置

在 CSS 规则定义对话框的“扩展”选项面板中定义扩展样式，如图 4.34 所示。

(1) 分页：通过样式来为网页添加分页符号。允许用户指定在某元素前或后进行分页，分页是指打印网页中的内容时在某个指定的位置停止，然后将接下来的内容打在下一张纸上。

(2) Cursor(光标)：改变光标形状，光标放置于此设置修饰的区域上时，形状会发生改变。

(3) Filter(过滤器)：使用 CSS 语言实现的滤镜效果，在下拉列表中有多种滤镜可供选择。



图 4.34 “扩展”选项面板

4.4 编辑 CSS 样式

定义过 CSS 样式后,还可以对 CSS 样式进行编辑和修改。

4.4.1 修改 CSS 样式

修改 CSS 样式的具体步骤如下。

(1) 在“CSS 样式”面板中选中要修改的样式。

(2) 单击“CSS 样式”面板右下角的“编辑样式表”图标,打开样式的“CSS 规则定义”对话框。

(3) 在样式的“CSS 规则”对话框中重新定义 CSS 样式即可。

4.4.2 删除 CSS 样式

删除 CSS 样式的步骤如下。

(1) 在“CSS 样式”面板中选中要删除的样式。

(2) 单击“CSS 样式”面板右下角的“删除 CSS 样式”图标;或在“CSS 样式”面板中选中样式并单击右键,从弹出的快捷菜单中选中“删除”命令。

(3) 这时样式即被删除,同时从 CSS 规则中消失。

4.4.3 复制 CSS 样式

复制 CSS 样式的具体步骤如下。

(1) 在“CSS 样式”面板中选中要复制的样式。然后单击右键,在弹出的快捷菜单中选中“复制”命令。

(2) 打开"复制 CSS 规则"对话框,从中可以重新命名及修改样式。

(3) 设置完成后单击"确定"按钮,即可完成 CSS 样式的复制操作。

4.5 网页布局规划

4.5.1 表格的基础知识

使用表格布局页面是网页布局最基本的方法。由于浏览器与表格均为矩形形状,因此使用表格分割画面是最简单的方式。使用表格排版的页面在不同平台,不同分辨率的浏览器中都能保持其原有的布局,且在不同的浏览器平台中具有较好的兼容性。

1. 插入表格

表格是由行和列组成的,并根据行和列的个数决定形状。行和列交叉形成了矩形区域,即表格中的一个矩形单元称为单元格。在 Dreamweaver 中利用"表格"对话框可以插入表格。执行"插入"|"表格"命令或在"插入"面板的"常用"分类中单击"表格"按钮,打开"表格"对话框,如图 4.35 所示。



图 4.35 "表格"对话框

(1) "行数/列":指定表格的行和列的个数。

(2) "表格宽度":指定将表格宽度以像素单位或浏览器窗口宽度为基准的百分比(%)单位。

(3) "边框粗细":用像素单位来指定表格边框线的厚度。如果不想显示表格的边框线,可以输入 0。

(4) "单元格边距":指定单元格的内容以及单元格边框之间的空白。不输入具体数值时,默认值为 1 像素。

（5）“单元格间距”：指定单元格之间的空白。不输入具体数值时，默认值为 2 像素。

（6）“标题”：将表格的一行或一列表示为表头时，选择所需的标题样式。

（7）“辅助功能”：指定针对表格设置的辅助选项。

①“标题”：指定表格的标题。

②“摘要”：输入关于表格的摘要说明。该内容虽然不显示在浏览器中，但可以在屏幕阅读器上识别，并可以转换为语音。

插入表格后，将会在表格的最下方出现整个表格和各列的列宽值。调节表格宽度时，该值也会一起变化，因此很容易调节表格。

有时会同时出现两种数字，这是由于 HTML 代码中设置的列宽度和实际页面中显示的列宽度不一致造成的。在这种情况下，单击表示整个表格宽度的数字，在弹出的下拉列表中选择“使所有宽度”选项，即可使代码与页面中显示的宽度一致。

2. 调整表格

插入表格后，可以通过调节表格大小等操作来制作所需的形状。当表格周围出现黑色边框，就表示已经选择了该表格。将光标移动到尺寸手柄上，光标会变成←→或 ↕ 形状。在此状态下按住鼠标左键，向左右、上下或对角线方向上拖动就可以调节表格的大小。当光标移动到表格右下方的手柄处，光标变成 ⤡ 形状时，可以向下拖动鼠标来增大表格高度。插入表格后，在操作过程中可能会出现表格的中间需要嵌入单元格或删除不需要的单元格的情况。此时，执行 Dreamweaver CS6 中的添加和删除表格的相关命令即可。添加行或列的时候，可以添加一行或一列，也可以同时添加多个行或列。

1）添加行和列

在插入的表格中添加行或列时，右击表格，在弹出的快捷菜单栏中执行“表格”|“插入行（列、行或列）”命令，可以更加快捷地进行操作。执行“插入行或列”命令后，将弹出“插入行或列”对话框，如图 4.36 所示。在该对话框中设置行数、列数以及插入位置。



图 4.36　“插入行或列”对话框

（1）“插入”：选择添加“行”或添加“列”。

（2）“行（列）数”：输入要添加的行或列的个数。

（3）“位置”：选择添加行或列的位置。

技巧：将插入点置于表格最后一行的最后一个单元格中，按 Tab 键，可快速插入一个新的行。

2）删除行和列

删除行或列的最简单的方法是选择想删除的行或列后，按 Delete 键。例如，单击第一列的第一个单元格，按住鼠标左键并拖动到第一列的最后一个单元格，即选中该列，按 Delete 键即可。或者选中要删除的行或列，单击鼠标右键，在弹出的快捷菜单中执行“表格-删除行（删除列）”命令，即可删除选中的行或列。

用户还可以一次对多个单元格进行复制或粘贴操作，既可以选择在复制和粘贴时保留原格式，也可以选择在复制和粘贴时仅操作单元格中的内容。

3. 编辑表格

1) 设置表格属性

在页面中选择表格后，在文档窗口下方将会显示表格的“属性”面板，如图 4.37 所示。在该面板中可以调整行和列的个数、表格的宽度和高度、单元格的空格等属性。表格“属性”面板中的各项参数含义如下。

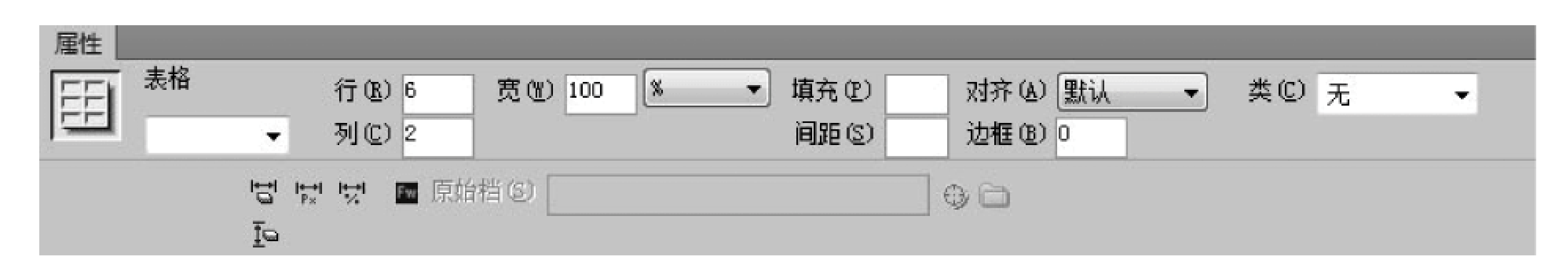


图 4.37 设置表格属性

(1) “表格”：输入表格名称。

(2) “行、列”：输入构成表格的行和列的个数。

(3) “宽”：指定表格的宽度。以当前文档的宽度为基准，可以指定百分比或像素为单位。默认单位为像素，若想固定大小，需继续使用像素单位。

(4) “填充”：设置单元格内容和单元格边框之间的间距。可以认为是单元格内侧的空白。将该值设置为 0 以外的数值时，在边框和内容之间会生成间隔。

(5) “对齐”：设置表格在文档中的位置。包括：“默认”、“左对齐”、“居中对齐”和“右对齐”4 个选项。

(6) “类”：设置表格样式。

(7) “间距”：设置单元格之间的间距。该值设置为 0 以外的数值时，在单元格和单元格之间会出现空格，因此两个单元格之间有一些间距。

(8) “边框”：设置表格的边框厚度。大部分浏览器中，表格的边框都会采用立体性的效果方式，但在整理网页文件而使用的布局表格中，最好不要显示边框。在这种情况下，就需要将“边框”的值设置为 0。

2) 设置单元格属性

在页面中选择单元格后，在文档窗口下方会显示单元格的“属性”面板，如图 4.38 所示。在该面板中可以设置单元格的背景颜色或背景图像、对齐方式、边框颜色等各种属性，还可以将一个单元格拆分成几个单元格或将多个单元格合并为一个单元格。单元格的“属性”面板中各项参数含义如下。



图 4.38 设置单元格属性

(1) 合并所选单元格、使用跨度：选择两个以上的单元格后，单击该按钮，就可以合并这些单元格。

（2）拆分单元格为行或列：单击该按钮后，选择行或列以及拆分的个数，就可以拆分所选单元格。

（3）垂直：设置单元格中的图像或文本的纵向位置。包括“顶端”、“居中”、“底部”、“基线”和“默认”5 种形式。

（4）水平：设置单元格中的图像或文本的横向位置。

（5）宽、高：设置单元格的宽度和高度。

（6）不换行：输入文本时，即使超出单元格的宽度，也不会换行。在不换行的情况下继续横向输入，就会增大单元格的宽度。

（7）标题：为了与其他内容区别，明显地表示单元格标题并居中对齐。

（8）背景颜色：指定单元格的背景颜色。

4.5.2　使用框架

框架指的是把一个画面分成几个区域，并在各个区域中显示不同网页文件的结构，利用框架可以更容易整理文档。框架的英文是“frame”，是指网页在一个浏览器窗口下分割成几个不同区域的形式。

1. 框架的概念

学习框架之前，首先要了解框架和框架集的关系。框架和框架集的用语类似，因此容易混淆。如图 4.39 所示，该文档将页面分成上、下两个大页面(其中下部分又分成左下和右下两个页面)，这样分成的各个画面称为框架。因此，该文档一共由两个框架组成。





图 4.39　框架和框架集显示效果

根据需要分解框架后，在各个框架上再插入已准备的文档。在各个框架上填充文档后会保存整体结构，而合并这两个画面的总体框架称为框架集。框架集中包括当前文档的框架组成个数、各个框架的大小、各框架的连接文件等相关信息。并且，在浏览器上只显示框架集文件，即可一次性显示出所有框架文件。

由两个框架形成的文档包括各个框架的相关文档和框架集文档，所以共包含三个文档。框架的作用就是把浏览器窗口划分为若干个区域，每个区域可以分别显示不同的网页。这样重复出现的内容被固定下来，每次浏览者发出对页面的请求时，只下载发生变化的框架的页面，其他子页面保持不变，必然就会给浏览者带来方便，节省时间。使用框架可以非常方便地完成导航工作，而且各个框架之间绝不存在互相干扰的问题，所以框架技术一直被广泛应用于页面导航。

可以在框架集中再插入其他框架集，这种做法通常称为“嵌套框架集”，而这些框架集也经常应用在由框架组成的网页中。

如图 4.40 所示，左下侧结构的框架集是先制作上下框架结构的框架集，并在下侧框架中再插入一个左右框架集所得到的框架集。

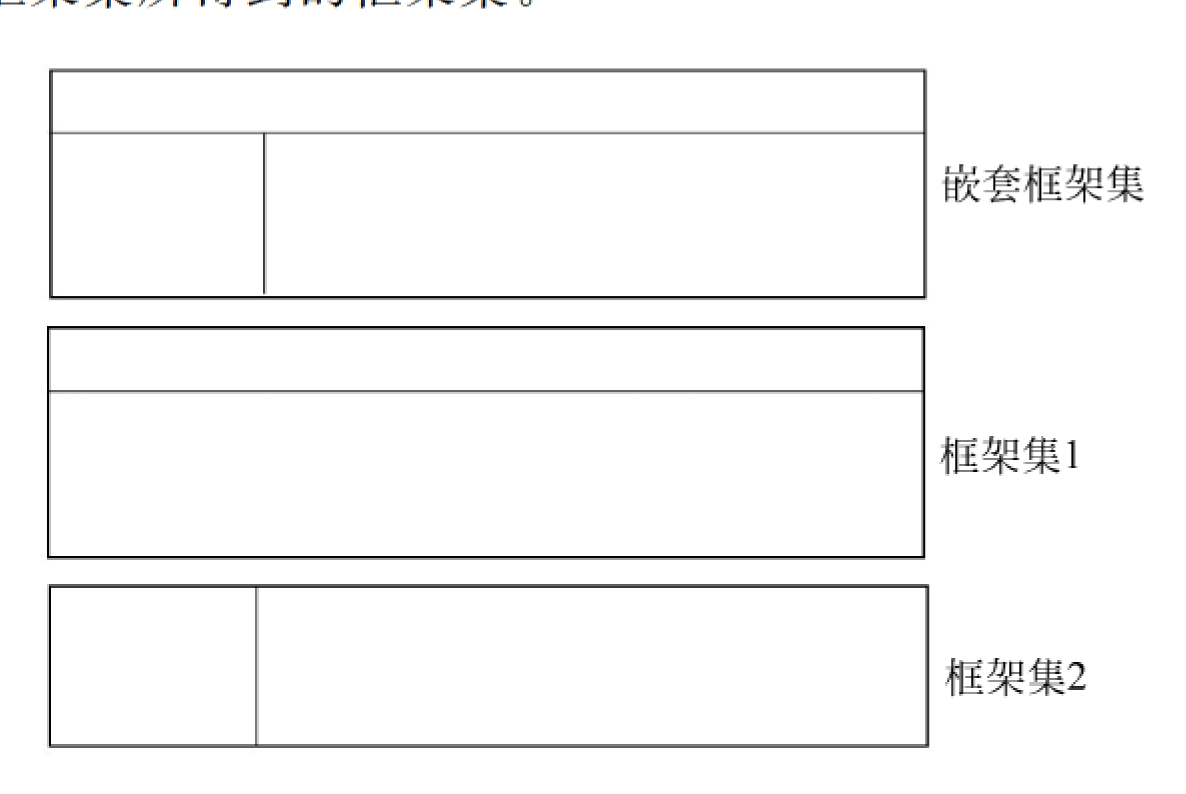


图 4.40　嵌套框架集

2. 设置框架集属性

在“框架”面板中单击外边框，就可以选中框架集并显示对应的“属性”面板，如图 4.41 所示。此时，在“属性”面板中出现如下框架集相关属性。

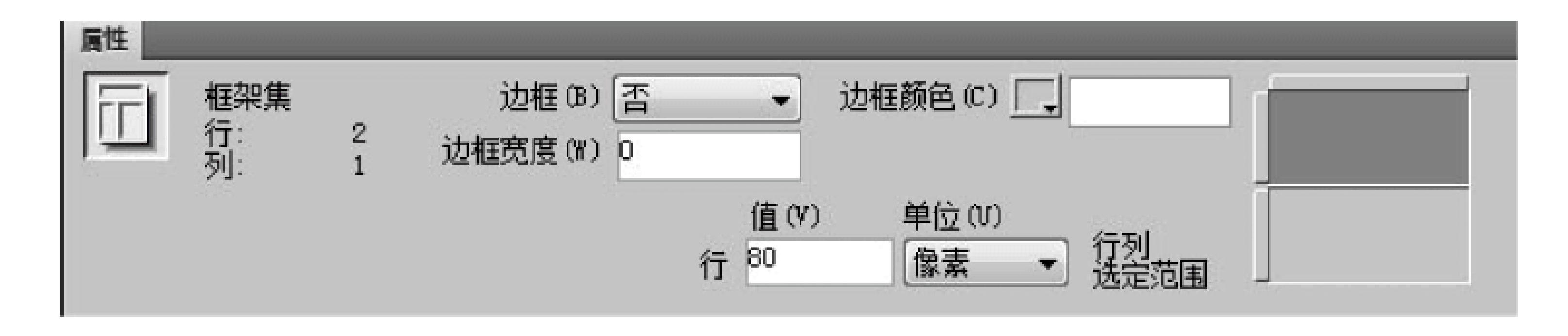


图 4.41　框架集“属性”面板

(1) “边框”：设置框架是否有边框。“是”选项为有边框，“否”选项为无边框，“默认”选项为浏览器决定是否有边框。对于大多数浏览器而言，默认为有边框，如果框架集属性的设置和框架属性的设置相冲突，以框架属性的设置为主。

(2)“边框颜色”：设置边框的颜色，可以单击颜色框，在拾色器面板中进行选择。

(3)“边框宽度”：设置框架结构中边框的宽度，单位是像素。

(4)设置框架结构的拆分比例：如果拆分的形式是上下拆分，那么显示“行”项目的数据；如果拆分的形式是左右拆分，则显示“列”项目的数据。在最右侧的框中，可以选择要进行设置的框架，选择后会在“值”和“单位”两项中出现该框架对应的值。“值”项目对于“行”来说指的是高度，对于“列”项目来说指的是宽度。“值”的取值和“单位”的设置有着密切的关系。“单位”共有三个选项可供选择。

① 像素：为框架的高或宽设置绝对值。如果不希望在用户调整屏幕分辨率或者浏览器窗口时框架大小发生改变，例如在框架中插入网页的标题、导航条等时，可以采用为框架设置绝对值的办法。但是，网页中的每个框架不可能都设置为像素值，有一个框架设置为像素值，通常会有其他框架的高度设置为相对值。

② 百分比：是指框架在其所在的框架结构中所占总高或总宽的百分比。

③ 相对：在其他框架设置了以像素或百分比为单位的高宽后，剩余的高宽会分配给单位设置为“相对”的框架。使用“相对”作为单位时，通常不需要设置值，有的时候为了保证跨浏览器时的兼容性，可以设置值为“1”。

3. 设置框架属性

在“框架”面板中选择框架时或在操作画面中选择框架时，在“属性”面板中出现如图4.42所示的框架相关属性。

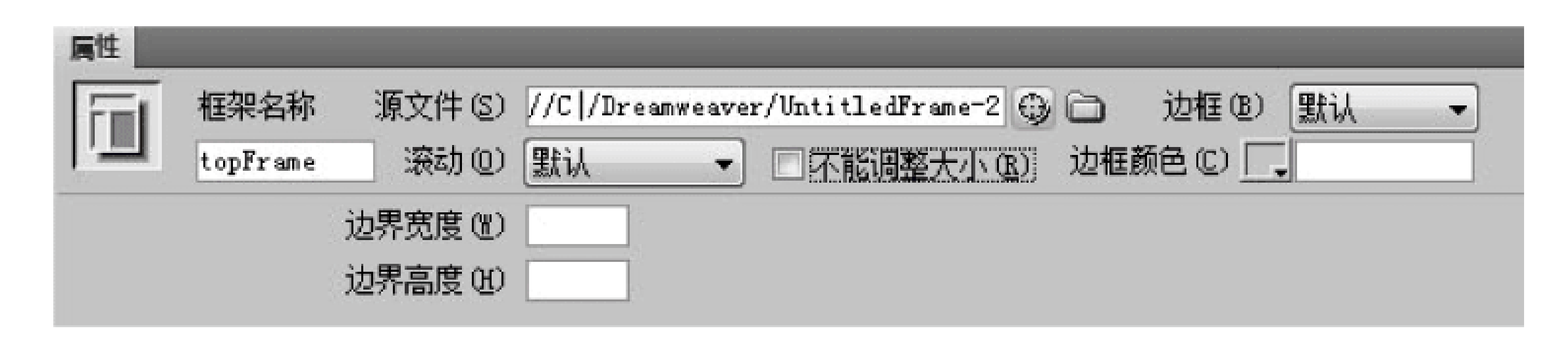


图4.42　框架“属性”面板

(1)“框架名称”：显示所选框架的名称(不要与文件名称混淆)。输入框架名称，才可以应用框架与其他文件链接等各种功能。框架名称必须以英文字母开头，且不可以使用特殊字符或空格。

(2)“源文件”：指定当前选择的框架中出现的文件名称。可以单击“浏览”按钮选择文件或直接输入文件所在的路径。

(3)“滚动”：选择在框架中是否显示滚动条，包含以下几个选项。

① 是：显示滚动条。

② 否：即使无法显示框架的一部分，也不显示滚动条。

③ 自动：内容比框架区域更长的时候，显示滚动条。

④ 默认：设置为默认值。

(4)“边框”：选择是否显示当前框架的边框。若选择“是”，显示边框；若选择“否”，不显示边框。

(5)“边界宽度/边界高度”：设置框架内容的周围边界。例如，设置10像素的边界宽度，会在左侧和右侧分别留出5像素的空白。

(6)“不能调整大小”：勾选此复选框即可固定框架的大小，使用户不能随意调节框架的大小。

(7)“边框颜色”：指定框架的边框颜色。

4. 设置框架链接

使用框架的最大理由是可以固定页面的一部分，而且可以只更改需要更改的框架的内容，即在不改变菜单所在框架的情况下，只变动菜单中的相关内容。若想在特定框架中显示所需的内容，就要区分多个不同的框架，因此，应该给各个框架间设置链接。

在 Dreamweaver CS6 中制作框架时，会自动生成框架名称，用户可以按需要适当更改框架名称。但框架名称一定要输入英文，而且不能使用特殊字符或空格。输入框架名称后，在“属性”面板的“目标”下拉列表中选择所需目标，即表示内容的框架，如图 4.43 所示。



图 4.43 框架“属性”面板中的“目标”

“属性”面板中的“目标”下拉列表中不仅包括名称，而且还提供_blank、_parent、_self、_top 等属性选项。应用链接方式来链接当前文档和其他文档时，最好在“目标”下拉列表中选择_blank 或_top 选项。

(1) _blank：在新窗口中显示链接内容。

(2) _parent：链接内容出现在包括当前框架文档的父框架集上，大部分都会出现与应用_top 选项一样的结果。

(3) _self：链接内容出现在应用链接的窗口或框架中。在没有另外指定“目标”的情况下，会作为默认值。

(4) _top：删除所有当前框架结构，并显示链接内容。

5. 制作内联框架

框架文件的优点是在保持菜单等一部分内容的情况下，可以更换其中的实际内容，因此，比较容易维持网页的整体设计。但整个网页由一个图像组成时，很难利用框架结构，因为将切割图像插入到多个框架时，很难显示成一个图像。

将切割后的小图像最终显示为一个大图像的最好方法是利用表格。那么，在利用表格来制作的布局中，可不可以像框架文件一样，在固定一部分的同时只更改其中的实际内容呢？这种情况下，可以应用内联框架，内联框架也可以称为“浮动框架”。

如图 4.44 所示，网页中顶部和左侧部分图像是固定的，而右下侧部分为利用链接来链接起来的文档。这是利用表格来制作的布局，在显示实际内容的右侧部分单元格中插入了内联框架。

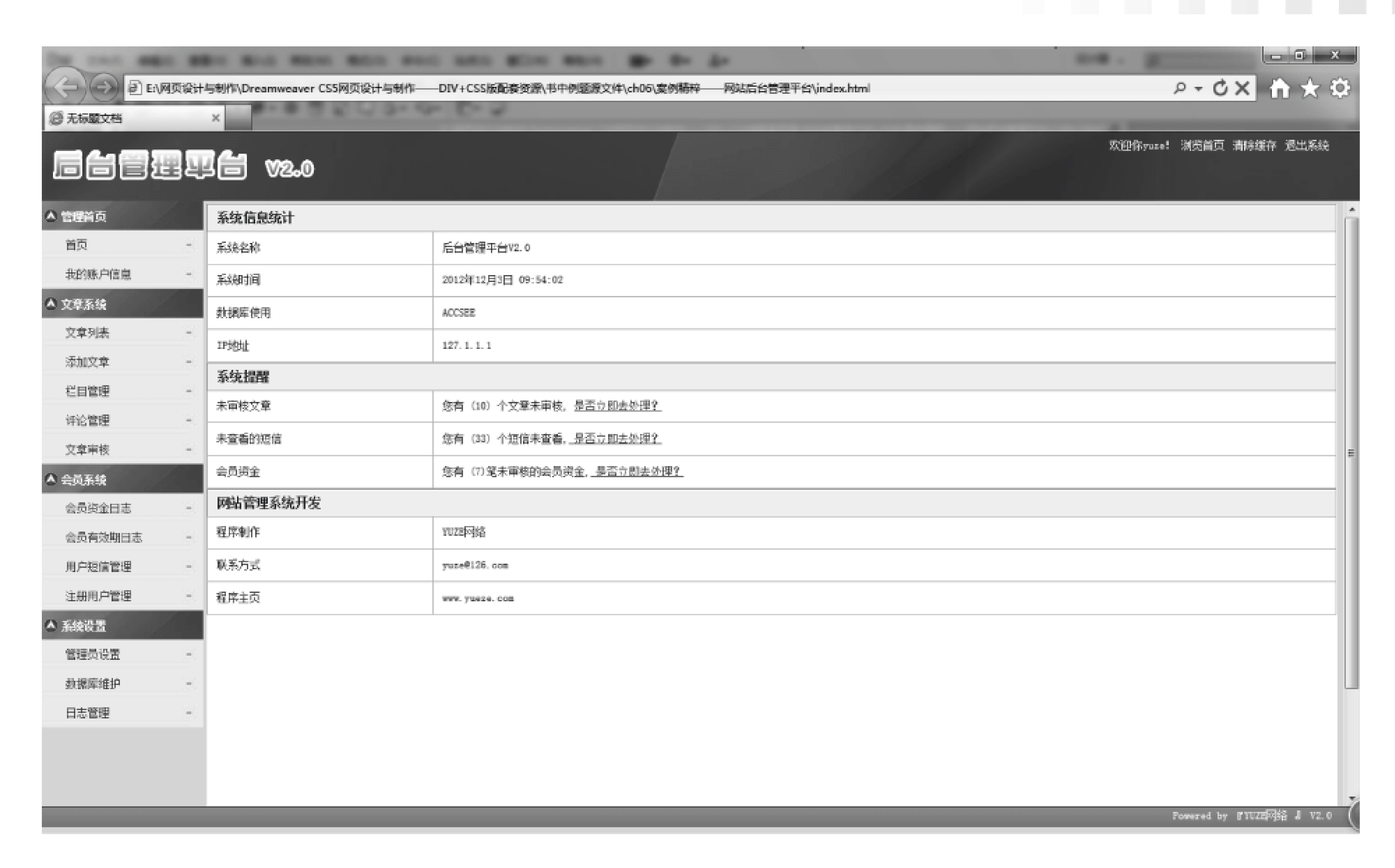


图 4.44　内联框架示意图

4.5.3　使用 CSS 布局网页

1. 初识 DIV

<div>标记作为一个容器标记被广泛地应用在 HTML 中。利用这个标记，加上 CSS 对齐控制，可以很方便地实现各种效果。

创建 DIV 的方法：<div>简单而言就是一个区块容器标记，即<div>与</div>之间相当于一个容器，可以容纳段落、标题、表格、图片，乃至章节、摘要和备注等各种 HTML 元素。因此，可以把<div>与</div>中的内容视为一个独立的对象，用于 CSS 的控制。声明时只需要对<div>进行相应的控制，其中的各种标记元素都会因此而改变。

【例 4.18】　<div>标记实例。

```
<html>
<head>
<title>div 标记范例</title>
<style type="text/css">
<!--
div{
        font-size:18px;
        font-weight:bold;
        font-family:Arial;
        color:#FF0000;
        background-color:#FFFF00;
        text-align:center;
        width:300px;
        height:100px;}
-->
</style>
```

```
</head>
<body>
<div>
这是一个 div 标记
</div>
</body>
</html>
```

在上面的实例中通过 CSS 对<div>块的控制，制作了一个宽 300 像素和高 100 像素的黄色区块，并进行了文字效果的相应设置，在浏览器中的执行结果如图 4.45 所示。



图 4.45　创建 DIV

2. CSS 定位与 DIV 布局

网页中的各种元素都必须有自己合理的位置，从而搭建出整个页面的结构。下面围绕 CSS 定位的几种原理方法，深入介绍使用 CSS 对页面中的块元素定位的方法。

1) 盒子模型

盒子模型是 CSS 控制页面时一个很重要的概念。对所有页面中的元素都可以看成是一个盒子，它占据着一定的页面空间。一般来说，这些被占据的空间往往都要比单纯的内容要大，换句话说，可以通过调整盒子的边距和距离等参数来调整盒子的位置。

一个盒子模型是由 content(内容)、border(边框)、padding(间隙)和 margin(间隔)等 4 个部分组成，如图 4.46 所示。

一个盒子的实际宽度(或高度)是由 content＋padding＋border＋margin 组成的。在 CSS 中可以通过设定 width 和 height 的值来控制 content 的大小，并且对于任何一个盒子，都可以分别设定 4 条边各自的 border、padding 和 margin。因此只要利用好盒子的这些属性，就能够实现各种各样的布局效果。

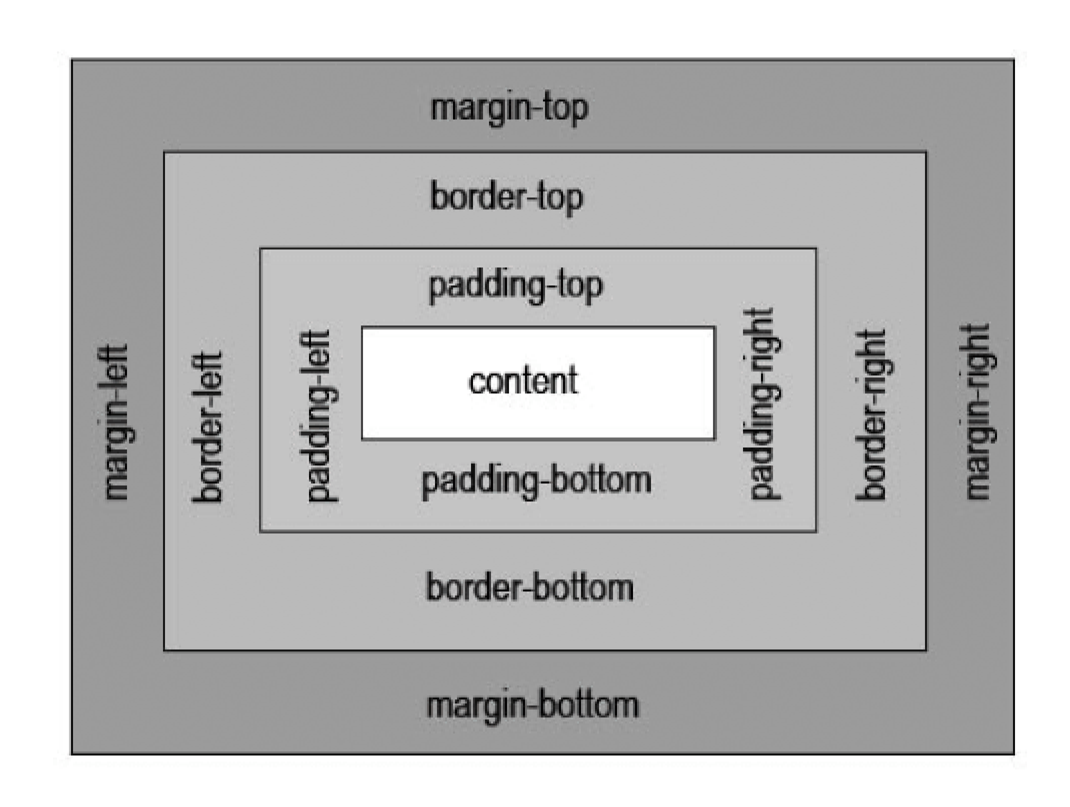


图 4.46 盒子模型

关于盒子模型,有以下几点要注意。

(1) 边界值可为负值,其显示效果各个浏览器可能不同。

(2) 边框默认的样式(border-style)为不显示(none)。

(3) 填充值不可为负。

(4) 对于块级元素,未浮动的垂直相邻元素的上边界和下边界会被压缩。例如,有上下两个元素,上面元素的下边界为 10px,下面元素的上边界为 5px,则实际两个元素的间距为 10px(两个边界值中较大的值)。

(5) 浮动元素(无论左或右浮动)边界不压缩,且若浮动元素不声明宽度,则其宽度趋向于 0,即压缩到其内容能承受的最小宽度。

(6) 内联元素,例如标签<a>,定义上下边界不会影响到行高。

(7) 如果盒子中没有内容,则即使定义了宽度和高度都为 100%,实际上只占 0,因此不会被显示,此点在采取布局的时候需要特别注意。

2) 元素的定位

网页中的各个元素都必须有自己合理的位置,从而搭建出整个页面的结构。围绕 CSS 定位有以下几个常用标签,包括 position、float 和 z-index 等。

(1) float 定位

float 定位是 CSS 排版中非常重要的手段,属性 float 的值很简单,可以设置为 left、right 或者默认值 none。当设置了元素向左或者向右浮动时,元素会向其父元素的左侧或右侧靠紧。

【例 4.19】 float 属性实例。

```
<html>
<head>
<title>float 属性</title>
<style type="text/css">
<!--
body{
    margin:15px;
    font-family:Arial;
    font-size:12px;
}
```

```
.father{
    background-color:#ff0000;
    border:1px solid #111111;
    padding:25px;
    /* 父块的 padding */
}
.son1{
    padding:10px;
    /* 子块 son1 的 padding */
    margin:5px;
    /* 子块 son1 的 margin */
    background-color:#ffff00;
    border:1px dashed #111111;
    float:left;                          /* 子块 son1 左浮动 */
}
.son2{
    padding:5px;
    margin:0px;
    background-color:#ffd270;
    border:1px dashed #111111;
}
-->
</style>
</head>

<body>
<div class="father">
<div class="son1">float1</div>
<div class="son2">float2</div>
</div>
</body>
</html>
```

没有设置块 son1 向左浮动前,页面效果如图 4.47 所示。



图 4.47　没有左浮动的效果

当设置了块 son1 的 float 值为 left 时，页面效果如图 4.48 所示。



图 4.48　设置左浮动的效果

（2）position 定位

position 定位与 float 一样，也是 CSS 排版中非常重要的概念。position 从字面意思上看就是指定块的位置，即块相对于其父块的位置和相对于它自身应该在的位置。

position 属性一共有 4 个值，分别是 static、absolute、relative 和 fixed。其中，static 为默认值，它表示块保持在原本应该在的位置上，即该值没有任何移动的效果。下面以 absolute 为例来解释 position 定位。

【例 4.20】　position 属性实例。

```
<html>
<head>
<title>position 属性</title>
<style type="text/css">
<!--
#father{
        background-color:#ffff66;
        border:1px dashed#000000;
        width:100%;
        height:100%;
        padding:5px;
        }
#block1{
       background-color:#fff0ac;
       border:1px dashed#000000;
       padding:10px;
       position:absolute;                    /* absolute 绝对定位 */
```

```
        left:30px;
        top: 35px;
        }
#block2{
        background-color:#ffbd76;
        border:1px dashed #000000;
        padding:10px;
        }

-->
</style>
</head>
</body>
      <div id="father">
           <div id="block1">absolute</div>
           <div id="block2">block2</div>
      </div>
</body>
</html>
```

上面的例子中将子块 1 的 position 属性值设置为 absolute,并且调整了它的位置,此时子块 2 便移动到了父块的最上端,如图 4.49 所示。此时子块 1 已经不再属于父块 #father,因为将其 position 值设置成了 absolute,因此子块 2 成为父块中的第一个子块,移动到了父块的最上方,如图 4.49 所示。

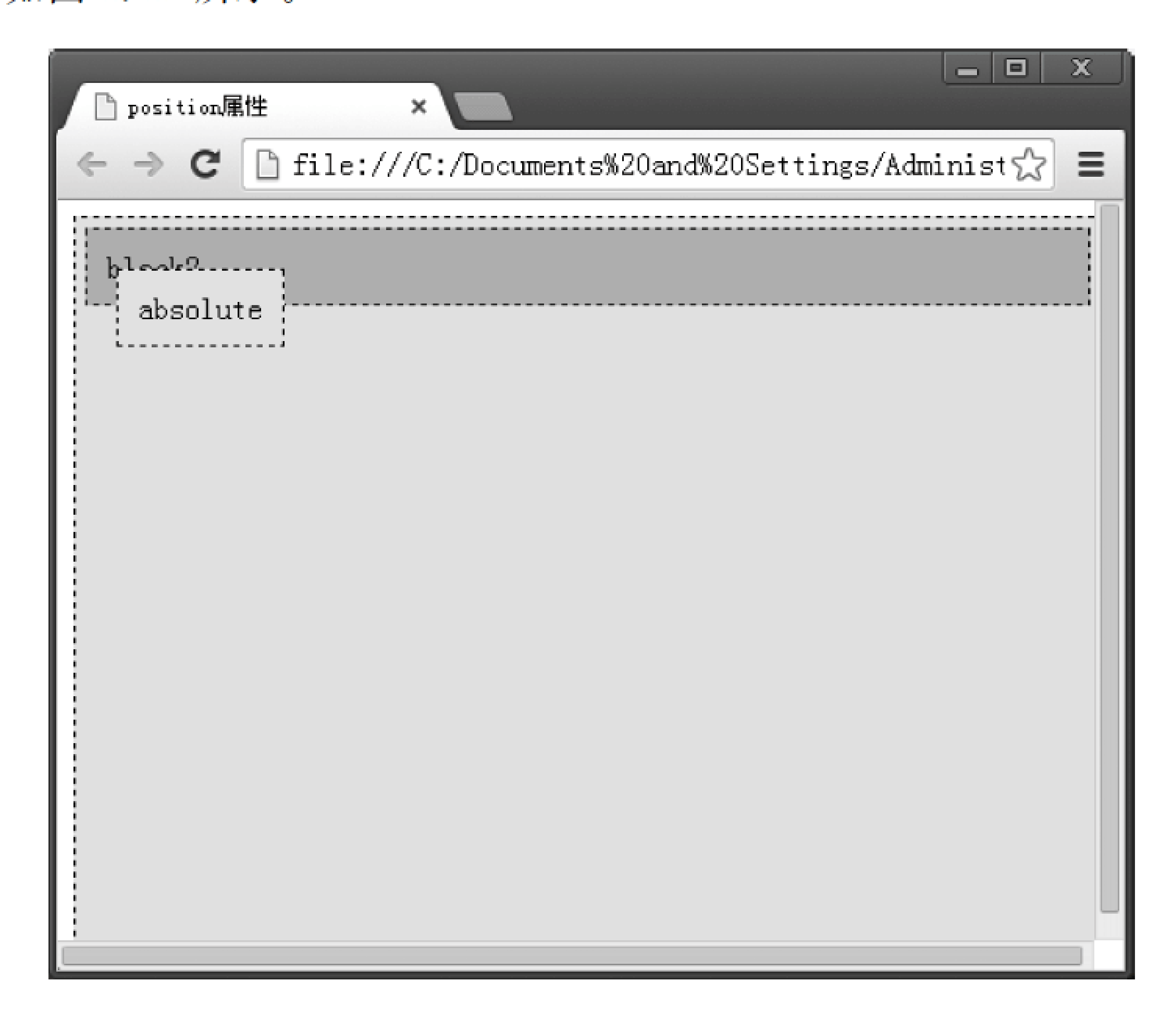


图 4.49 子块 1 的 position 属性为 absolute 设置效果

如果将两个子块的 position 属性同时设置为 absolute,这时两个子块都将不再属于其父块,都相对于页面定位。在上例的基础上进行修改,代码如下。

```
#block1{
        background-color:#fff0ac;
        border:1px dashed #000000;
        padding:10px;
        position:absolute;                    /*absolute 绝对定位*/
        left:30px;
        top: 35px;
        }
#block2{
        background-color:#ffbd76;
        border:1px dashed #000000;
        padding:10px;
         position:absolute;                   /*absolute 绝对定位*/
         left:50px;
         top:60px;
        }
```

当两个子块的 position 属性都设置为 absolute 时，它们都按照各自的属性进行定位，都不再属于其父块。两个子块有重叠的部分，且块 2 位于块 1 的上方，如图 4.50 所示。

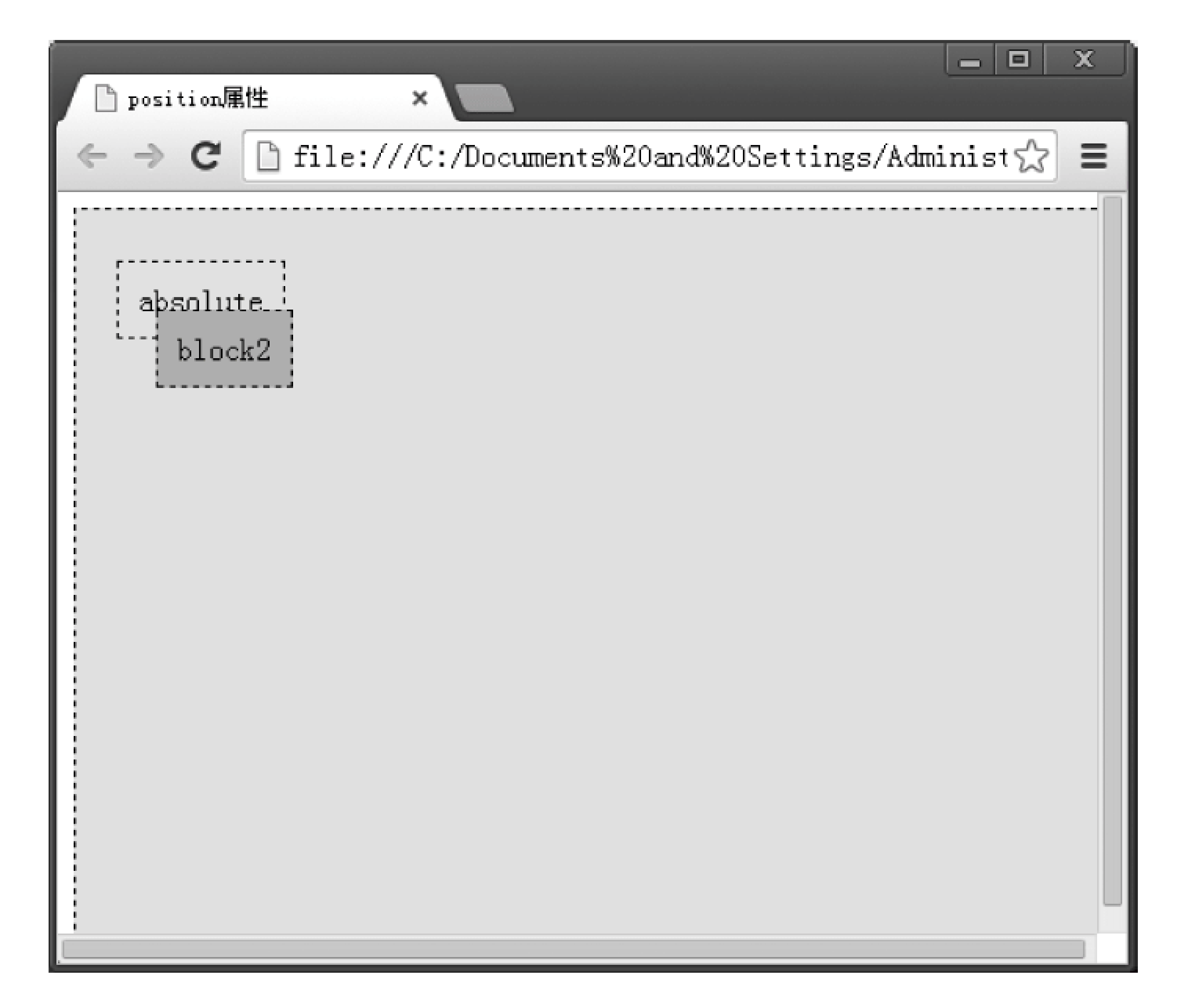


图 4.50　两个子块 position 属性都为 absolute 设置效果

(3) z-index 空间位置

z-index 属性用于调整定位时重叠块的上下位置，与它的名称一样，页面为 x-y 轴，垂直于页面的方向为 z 轴，z-index 值大的页面位于其值小的上方。

z-index 属性的值为整数，可以是正数也可以是负数。当块被设置了 position 属性时，该值便可设置各块之间的重叠高低关系。默认的 z-index 值为 0 时，当两个块的 z-index 值一样时，将保持原有的高低覆盖关系。

【例 4.21】 z-index 属性实例。

```
<html>
<head>
<title>z-index 属性</title>
<style type="text/css">
<!--
body{
    margin:10px;
    font-family:Arial;
    font-size:13px;
    }
#block1{
        background-color:#ff0000;
        border:1px dashed#000000;
        padding:10px;
        position:absolute;
        left:20px;
        top:30px;
        z-index:1;                          /*高低值1*/
        }
#block2{
        background-color:#ffc24c;
        border:1px dashed#000000;
        padding:10px;
        position:absolute;
        left:40px;
        top:50px;
        z-index:0;                          /*高低值0*/
        }
#block3{
        background-color:#c7ff9d;
        border:1px dashed#000000;
        padding:10px;
        position:absolute;
        left:60px;
        top:70px;
        z-index:-1;                         /*高低值-1*/
    }
-->
</style>
</head>
<body>
      <div id="block1">AAAAAAAAAA</div>
      <div id="block2">BBBBBBBBBB</div>
      <div id="block3">CCCCCCCCCC</div>
</body>
</html>
```

在上面的例子中对三个有重叠关系的块分别设置了 z-index 的值，设置前与设置后的效果分别如图 4.51 和图 4.52 所示。

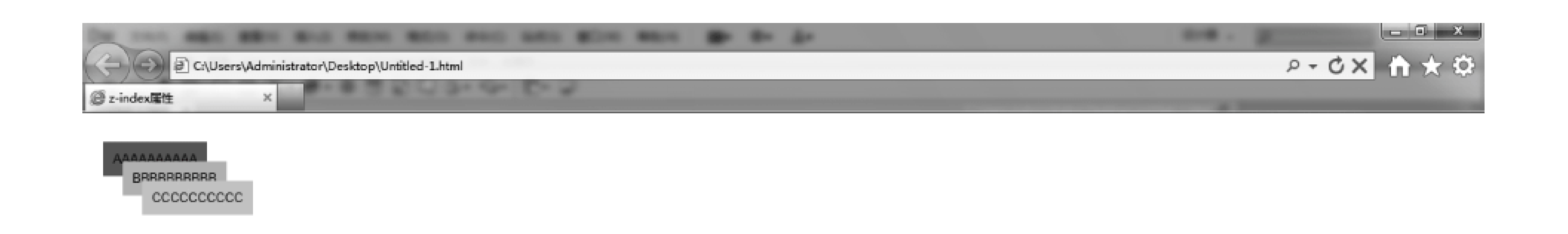


图 4.51　设置 z-index 前的效果



图 4.52　设置 z-index 后的效果

【例 4.22】 使用元素的定位方法实现文字的阴影效果。

```
<html>
<head>
<title>文字阴影效果</title>
<style type="text/css">
<!--
body{
```

```
    margin:15px;
    font-family:黑体;
    font-size:60px;
    font-weight:bold;
}
#block1{
        position:relative;
        z-index:1;
}
#block2{
        color:#AAAAAA;
/*阴影颜色*/
        position:relative;
        top:-1.06em;
/*移动阴影*/
        left:0.1em;
        z-index:0;
/*阴影重叠关系*/
}
-->
</style>
</head>
<body>
<div id="father">
        <div id="block1">定位阴影效果</div>
        <div id="block2">定位阴影效果</div>
</div>
</body>
</html>
```

效果如图 4.53 所示。



定位阴影效果

图 4.53　定位阴影效果

3. CSS+DIV 布局的常用方法

使用 DIV 可以将页面首先在整体上进行<div>标记的分块，然后对各个块进行 CSS 定位，最后再在各个块中添加相应的内容。这样进行<div>标记过的页面更新起来会十分容易，同时也可以通过修改 CSS 的属性来重新定位。

CSS 布局要求设计者首先对页面有一个整体的框架规划，包括整个页面分为哪些模块、各个模块之间的父子关系如何等。以最简单的框架为例，页面是由 banner、主体内容(content)、菜单导航(links)和脚注(footer)等几部分组成的，各个部分分别用自己的 id 来标识，整体内容如图 4.54 所示。

图 4.54 中的每个色块都是一个<div>，这里直接用 CSS 的 ID 标识方法来表示各个块。页面中的所有 DIV 块都属于块#container，一般的 DIV 布局都会在最外面加上一个父 DIV，以便对页面的整体进行调整。对于每个子块 DIV，还可以再加入各种块元素或者行内元素。

1) 设计各个块的位置

当页面的内容已经确定后，则需要根据内容本身来考虑整体的页面版型，例如单栏、双栏或左中右等。这里考虑到导航条的易用性，采用了常见的双栏模式，如图 4.55 所示。

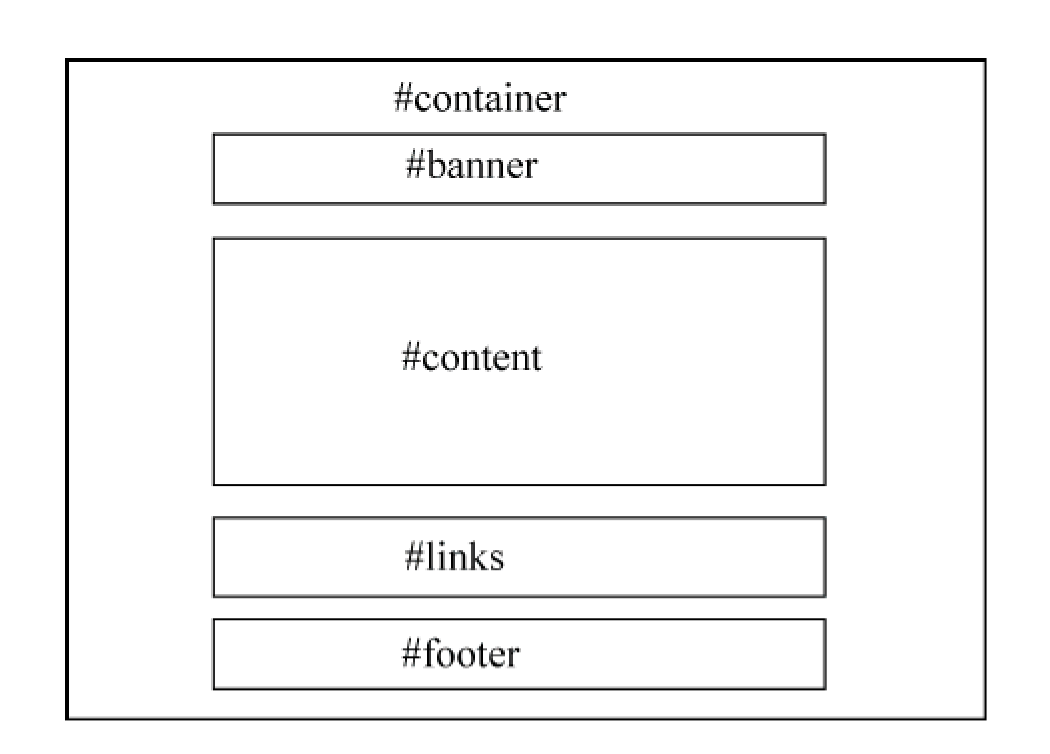


图 4.54　页面框架规划示意图

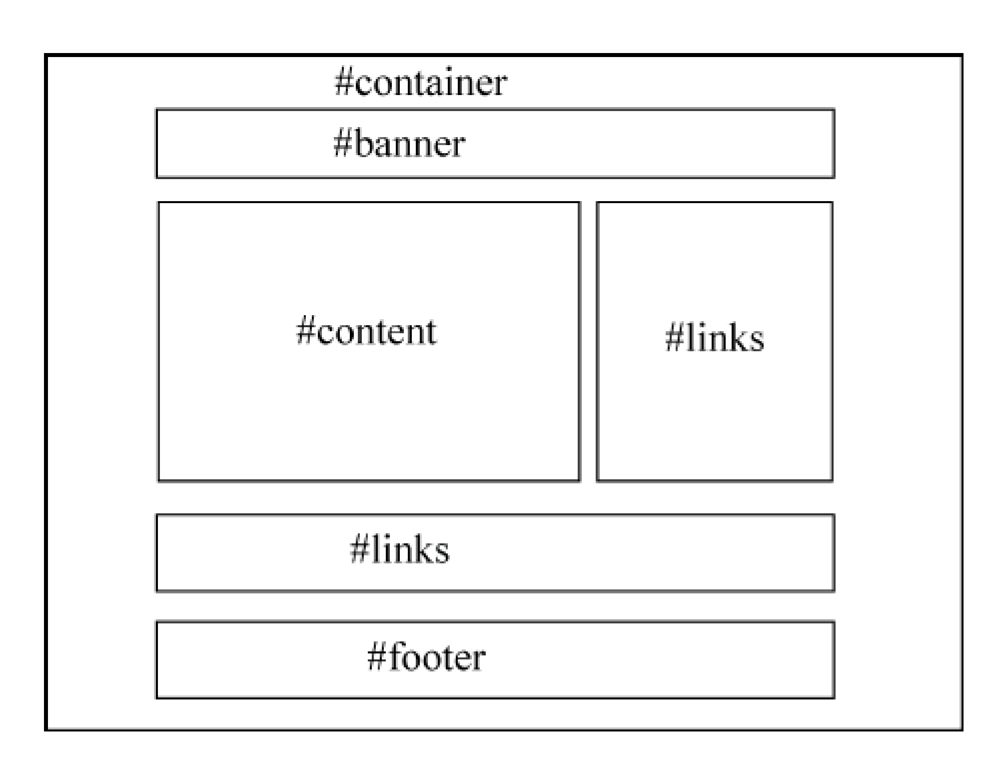


图 4.55　双栏模式示意图

在整体的#container 框架中页面的 banner 在最上方，然后是内容#content 与导航条#links，二者在页面的中部，其中#content 占据整个页面的主体。最下方是页面的脚注#footer，用于显示版权信息和注册日期等。有了页面的整体框架后，就可以使用 CSS 对各个 DIV 块进行定位了。

2) 使用 CSS 定位

整理好页面的框架后便可以利用 CSS 对各个块进行定位，实现对页面的整体规划，然后再往各个模块中添加内容。首先对<body>标记与#container 父块进行设置，代码如下。

```
body{
    margin:0px;
    font-size:13px;
    font-family:Arial, Helvetica, sans-serif;
```

```
}
#container{
    position:relative;
    width:100%;
}
```

以上设置了页面文字的字号、字体以及父块的宽度，让其撑满整个浏览器。接下来设置#banner块，代码如下。

```
#banner{
    height:80px;
    border:1px solid#000000;
    text-align:center;
    background-color:#a2d9ff;
    padding:10px;
    margin-bottom:2px;
}
```

这里设置了#banner块的高度，以及一些其他的个性化设置，当然也可以根据自己的需要进行调整。如果#banner本身就是一幅图片，那么对#banner的高度就不需要设置了。

利用float浮动方式将#content移动到左侧，#links移动到页面的右侧。这里不指定#content的宽度，因为它需要根据浏览器的变化而自己调整，但#links作为导航条则指定其宽度为200px。代码如下。

```
#content{
    float:left;
}
#links{
    float:right;
    width:200px;
    text-align:center;
}
```

在分别设置了#content和#links的浮动属性后，页面的块并没有按照想象的那样进行移动，#links被挤到了#content的下方，这是因为对#content没有设置宽度，它的宽度仍然是整个页面的100%。页面又需要占满浏览器的100%，因此不能设置#content的宽度，此时的解决办法就是将#links的margin-left设为负数，强行往左拉回200px，代码如下。

```
#links{
    float:right;
    width:200px;
    border:1px solid#000000;
    margin-left:-200px;                    /*往左拉回200px*/
    text-align:center;
}
```

此时会发现#content的内容与#links的内容发生了重叠，这时只需要设置#content

的 padding-right 为－200px，在宽度不变的情况下将内容往左挤回去即可。另外，由于＃content 和＃links 都设置了浮动属性，因此对＃footer 需要设置 clear 属性，使其不受浮动的影响。代码如下。

```
#content{
    float:left;
    text-align:center;
    padding-right:200px;
}
#footer{
    clear:both;
    text-align:center;
    height:30px;
    border:1px solid#000000;
}
```

这样，页面的整体框架便搭建好了。这里需要指出的是：＃content 中不能放宽度太长的元素，否则＃links 将再次被挤到＃content 的下方。

4. 常见版式布局实战

CSS 排版是一种全新的排版理念，它将页面首先在整体上进行<div>标记的分块，然后对各个块进行 CSS 定位，最后再往每个块里添加相应的内容。通过 CSS 排版的页面，更新容易，只需修改 CSS 属性来重新定位即可。

1）固定宽度且居中的版式

首先将所有的页面内容用一个大的<div>包围起来，代码如下。

```
<html>
<head>
<title>固定宽度且居中的版式</title>
</head>
<body>
<div id="container">页面具体内容</div>
</body>
</html>
```

指定该<div>的 ID 为 container，这个 ID 在整个页面中是唯一的。虽然大部分浏览器并不限制重复 ID 的使用，但因为会使得 JavaScript 等脚本语言在寻找对象时发生混乱，所以不建议在同一个页面中出现重复的 ID。

下面假设固定的宽度为 700px，介绍设置固定宽度且居中的版式的方法。代码如下。

```
<head>
<title>固定宽度且居中的版式</title>
<style>
body,html{
    margin:0;
    text-align:center;
}
#container{
```

```
    position:relative;
    margin:0 auto;
    width:700px;
    text - align:left;
}
</style>
</head>
```

上面这段代码中，首先对<body>和<html>标记进行属性控制，虽然90%以上的浏览器都是以<body>为基准的，但考虑到个别情况，因此二者同时声明，一般情况下不需要声明<html>标记。

第5行代码“margin：0；”指定页面四周的空隙都为0。紧接着设置“text-align：center；”，这是整个排版的关键语句，即将页面<body>中的所有元素都设置为居中，块#container属于页面的一部分，自然也居中对齐。

接下来设置#container的属性，“position：relative；”设置块相对于原来的位置。但是由于<body>已经居中，因此这里不需要再调整，只是考虑到浏览器的兼容性，所以加上了这句代码。

#container属性中的“margin：0；auto；”使得该块与页面的上下边界距离为0，左右边则自动调整。

然后设定“margin：0 auto；”，这是设定固定宽度。最后的“text-align：left；”用来覆盖<body>中设置的对齐方式，使得#container中的所有内容恢复左对齐。

2）左中右版式

将页面分割为左中右三块也是网页中常见的一种排版模式。下面制作的是左栏#left与右栏#right的固定宽度、位置，中间的#middle随着页面自动调整的布局方式。这样HTML的结构框架就已经清晰明了，下面直接用三个<div>块即可。代码如下。

```
<body>
<div id="left">
    <p>left</p>
</div>
<div id="middle">
    <p>正文内容</p>
</div>
<div id="right">
    <p>right</p>
</div>
</body>
```

设置<body>标记的样式，包括margin、padding、字体、颜色和背景色等，这些对整体结构都没有太大的影响，代码如下。

```
body{
    margin:0px; padding:0px;
    font - family:arial;
    color:#060;
    background - color:#CCC;
}
```

然后分别设置#left、#middle和#right的样式，其中，#left与#right都采用绝对定位，并且固定块的宽度，而#middle块由于需要根据浏览器自动调整，因此不设置类似的属性。但由于将另外两个块的position属性设置为了absolute，此时#middle的实际宽度为100%，因此必须将它的margin-left和margin-right都设置为190px(左右块的宽度)，代码如下。

```
#left{
    position:absolute;
    top:0px;
    left:0px;
    margin:0px;
    background:#FFF;
    width:190px;                          /* 固定宽度 */
}
#middle{
    padding:10px;
    background:#FFF;
    margin:0px 190px 0px 190px;           /* 左右空 190px */
    margin-top:0px;
}
#right{
    position:absolute;
    top:0px;
    right:0px;
    margin:0px;
    background:#FFF;
    width:190px;                          /* 固定宽度 */
}
```

这样整个左中右的框架就搭建好了。这时只需要将#left、#middle和#right这三个块嵌套在一个父块中就可以实现了，代码如下。

```
<div id="banner"></div>
<div id="mainbox">
   <div id="left"></div>
   <div id="middle"></div>
   <div id="right"></div>
</div>
<div id="footer"></div>
```

3）电子相册型版式

CSS对电子相册的排版有两种模式：幻灯片模式和详细信息模式。下面重点介绍详细信息模式的电子相册的排版方法。

详细信息模式要求每幅照片的信息能够显示在相片的一侧，并且不再更改页面的HTML拓扑结构。在采用了CSS的DIV排版后，仅需要在幻灯片的基础上不再浮动即可，然后将相片的超链接设置为向左浮动，相片的信息不再隐藏，关键代码如下。

```
body{
   Margin:0.8em;
```

```
    Padding:0px;
}
div.poc{
    width:450px;height:160px;
    margin:6px;
    padding:0px;
}
div.pic.img{
    border:1px solid#82c3ff;
}
div.pic.a.tn{
    float:left;
}
```

这样,只需要简单地修改 CSS 文件即可。

小结

CSS 是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。它的引入就是为了使得 HTML 能更好地适应页面的美工设计。它以 HTML 为基础,提供了丰富的格式化功能。本章主要介绍了 CSS 的功能,基础语法,以及如何对字体、段落、图片、背景等应用 CSS 进行美化。重点介绍了实际网页开发中常见的几种布局方式如何用 CSS 来实现。

习题

1. 如何将外部 CSS 样式文件链接到某个网页中?
2. 实现图文混排的核心语句是什么?
3. 操作题:应用 CSS+DIV 完成以下操作。

(1) 画一个 778px×250px 的方框(ID 命名为 header),相对整个页面居中,背景颜色为#eeeeee,边框 1px,实心,颜色为红色。

(2) 再画一个 778px×600px 的方框(ID 命名为 center),相对整个页面居中,背景颜色为白色,边框为 1px,实心,颜色为灰色,距 header 方框间隔 15px。

(3) 在 header 方框内,再画两个方框,分两列。

第一个方框(id 命名为 header-left),左对齐,距左边框 10px,距上边框 20px,长为 300px,高为 200px,背景为蓝色;

第二个方框(id 命名为 header-right),右对齐,距右边框 10px,上边框 5px,长为 420px,高为 30px,背景为白色。

(4) 在 center 方框内,画三个方框,分三列左中右排列。

左方框 id 命名为 center-left,宽 150px,高 400px,距左、上边框各为 10px,背景随意;

中方框 id 命名为 center-center,宽 400px,高 400px,距左、上边框同样各为 10px,背景随意;

右方框 id 命名为 center-right，高 400px，距左、右、上边框同样为 10px，背景随意，高度自己计算。

（5）在 header-right 中用 ul 和 li 的形式写出一组链接 5 个 li 排成一行，不允许换行，每个 li 的文字居中，li 前面不允许出现实心圆点或其他任何标记；每个 li 的长度为 70px，链接的文字内容：链接一、链接二、链接三、链接四、链接五（li 的个数是 5 个，链接目标为空链接）。

（6）定义整个网页的字号为 12px，文字颜色为＃666666，文字链接的颜色同样为＃666666，12px 大小，鼠标移上去的颜色为红色，移除时恢复为＃666666。

第5章 Dreamweaver的高级应用

本章要点：

- 行为；
- Spry 效果；
- JavaScript 介绍。

Dreamweaver 内置了多种行为以及可重用构件，可以方便地在页面中添加行为、插入用标准 HTML、CSS 和 JavaScript 编写的 Spry 构件，设置构件的样式，这样可以实现更加丰富的用户交互效果。

5.1 行为

优秀的网站，不仅包含文字和图像，还有很多交互式的效果，这种效果可以通过 Dreamweaver 中的一项强大的功能——行为来实现。Dreamweaver 内置了多种行为，即使用户不熟悉 JavaScript 代码的书写，也可以实现丰富的动态页面效果，达到用户与页面的交互。

5.1.1 认识行为

Dreamweaver 中的行为是一种运行在浏览器中的 JavaScript 代码，设计者可以将其设置在网页文档中，以允许浏览器与网页本身进行交互，从而以多种方式更改页面或引起某次任务的执行。

行为是由 Event 事件和 Action 动作两个基本元素组成的，通常，动作是一段 JavaScript 代码，利用这些代码可以完成相应的任务；而事件实际上是浏览器生成的消息，即事件由浏览器定义、产生和执行，指示该页面中在浏览时执行某种操作，这样事件指明执行某项动作的条件，例如，onMouseOver（鼠标经过）、onMouseOut（鼠标离开）、onClick（鼠标单击）等都是事件，当浏览者将鼠标指针移动到某个对象上时，浏览器为该对象生成一个 onMouseOver 事件，然后浏览器查看是否存在为对象在该事件发生时应该调用的 JavaScript 代码。每个页面元素所能发生的事件不尽相同，例如，页面文档本身能发生的 onLoad 事件和 onUnload 事件，分别对应页面被打开的事件和页面被关闭的事件。

从以上内容可以得出结论，在将行为附加到对象后，只要对该对象发生了所指定的事件，浏览器就会调用与该事件关联的动作。

在 Dreamweaver CS6 中，进行附加行为和编辑行为的操作都将用到“行为”面板。执行“窗口”|“行为”命令，打开“标签检查器”面板并自动切换到“行为”选项卡中，如图 5.1 所示。如果需要进行附加行为的操作，可以单击“行为”面板中的“添加行为”按钮 +，如图 5.2 所示，从弹出的菜单中选择需要添加的行为。

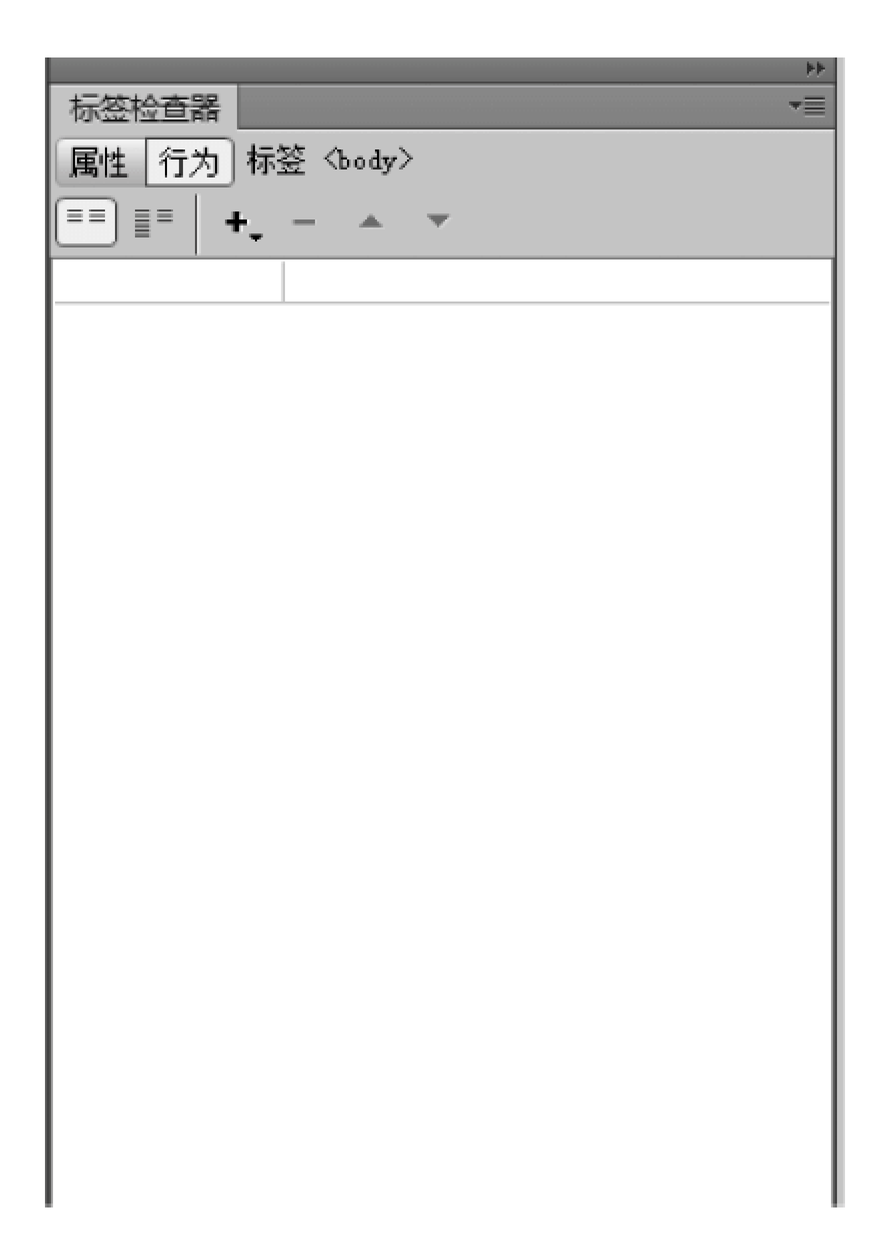



图 5.1　“行为”面板

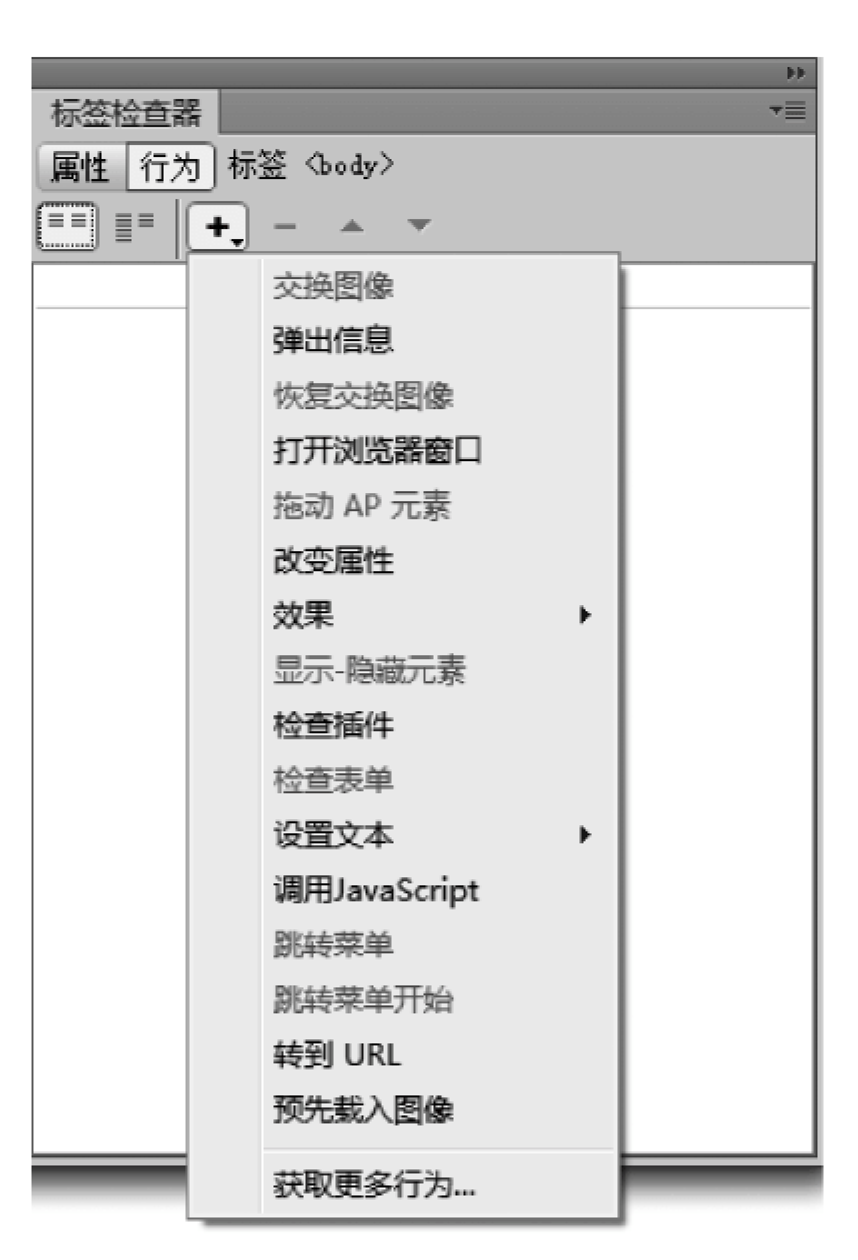


图 5.2　预设的各种行为

在“行为”面板上的列表中选择一个行为，单击该项左侧的事件栏，将显示一个下拉菜单，如图 5.3 所示。菜单中列出了所选行为中可用的触发事件，可根据实际需要来选择。



图 5.3　事件下拉列表

5.1.2　向页面添加行为

添加行为就是将行为添加到网页中的各个对象中，从而达到交互的效果。行为可以添加到整个文档中，也可附加到链接、图像、表单元素或其他 HTML 元素中的任一种之上。

在为网页添加行为时要按照以下三个步骤来执行：①选择对象；②添加动作；③设置触发事件。

如果当前网页已经附加了行为，那么行为将显示在“标签检查器”面板上的“行为”选项卡的列表中。如果需要删除网页中正在使用的行为，可以在列表中选择要删除的行为，单击“行为”选项卡中的“删除事件”按钮 - ，即可删除该行为。

1. 交换图像

“交换图像”行为的效果与鼠标经过图像的效果是一样的，该行为通过更改<img>标签中的 src 属性将一个图像与另一个图像进行交换。

步骤如下：

(1) 选中页面中需要添加“交换图像”行为的图像，如图 5.4 所示。



图 5.4 选中图像

(2) 单击“标签检查器”中的“添加行为”按钮，从弹出的菜单中选择“交换图像”选项，弹出“交换图像”对话框，设置如图 5.5 所示。单击“确定”按钮，完成“交换图像”对话框的设置，在“行为”面板中将自动添加相应的行为，如图 5.6 所示。

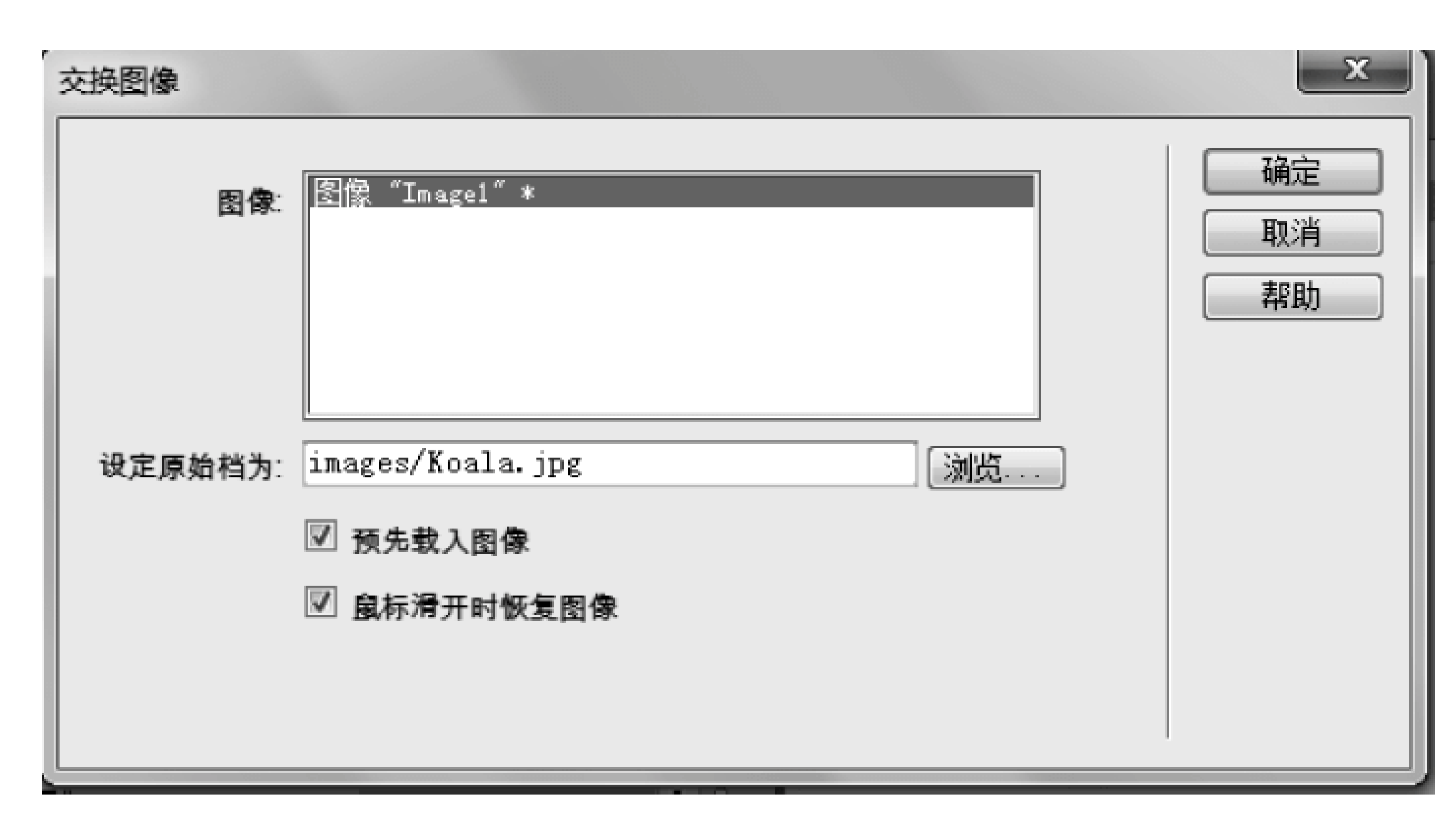


图 5.5 “交换图像”对话框

（3）保存页面，在浏览器中预览页面，当鼠标移至添加了“交换图像”行为的图像上时，可以看到交换图像的效果，如图 5.7 所示。

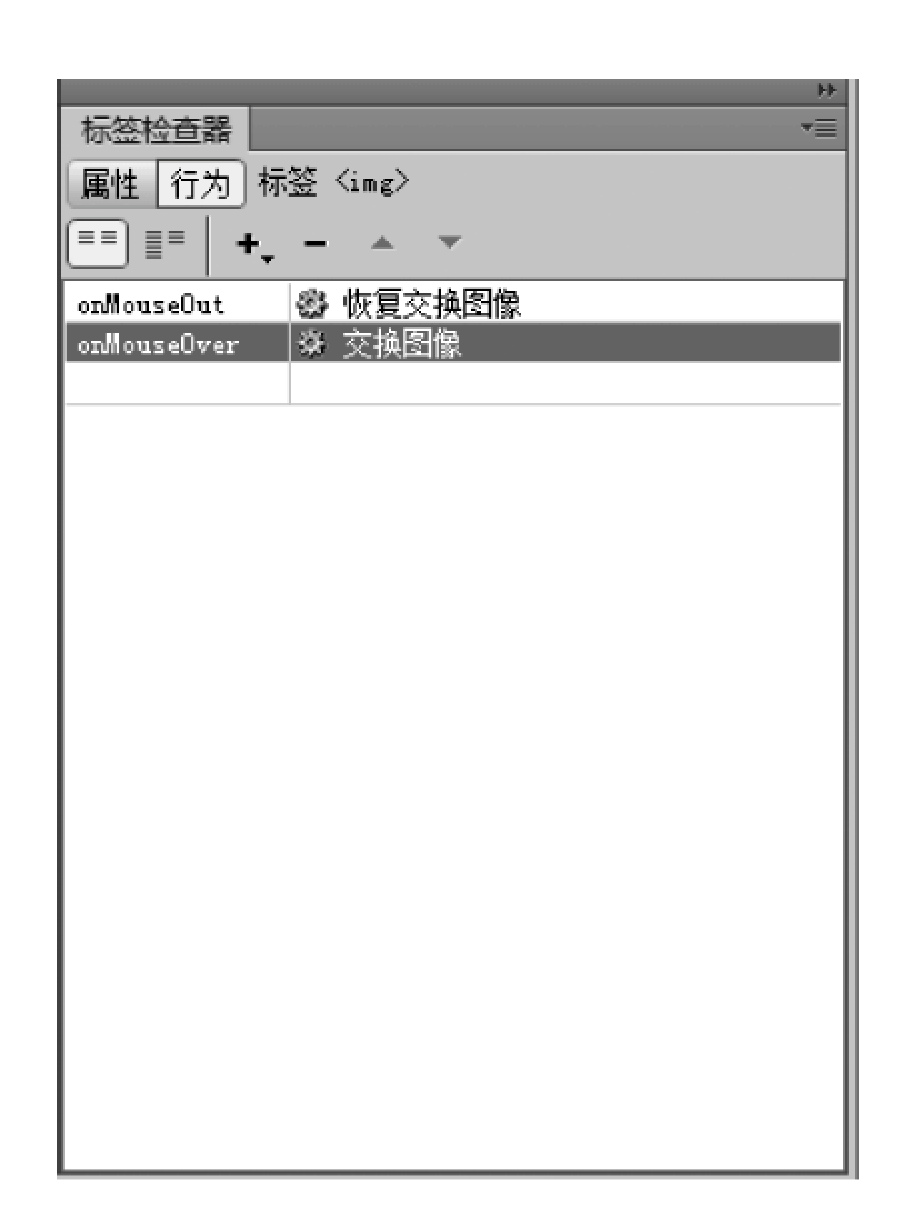



图 5.6　“行为”面板



图 5.7　预览“交换图像”行为效果

当在网页中添加“交换图像”行为时，会自动为页面添加“恢复交换图像”的行为，这两个行为的效果通常都是一起出现的。onMouseOver 触发事件表示当鼠标移至图像上时，onMouseOut 触发事件表示当鼠标移出图像上时。

2．设置状态栏文本

使用该行为可以使页面在浏览器左下方的状态栏上显示一些文本信息。如一般的提示链接内容、显示欢迎信息等，都可以通过这个行为的设置来实现。该动作可以加在 body 标签上，当页面载入后，状态栏显示文本。

步骤如下：

（1）在标签选择器中选中＜body＞标签，如图 5.8 所示。



图 5.8　选中＜body＞标签

（2）单击“行为”面板中的“添加行为”按钮 +，从弹出的菜单中依次选择“设置文本”|“设置状态栏文本”，弹出“设置状态栏文本”对话框，在“消息”文本框中输入显示在状态栏中的文本信息，如图 5.9 所示。

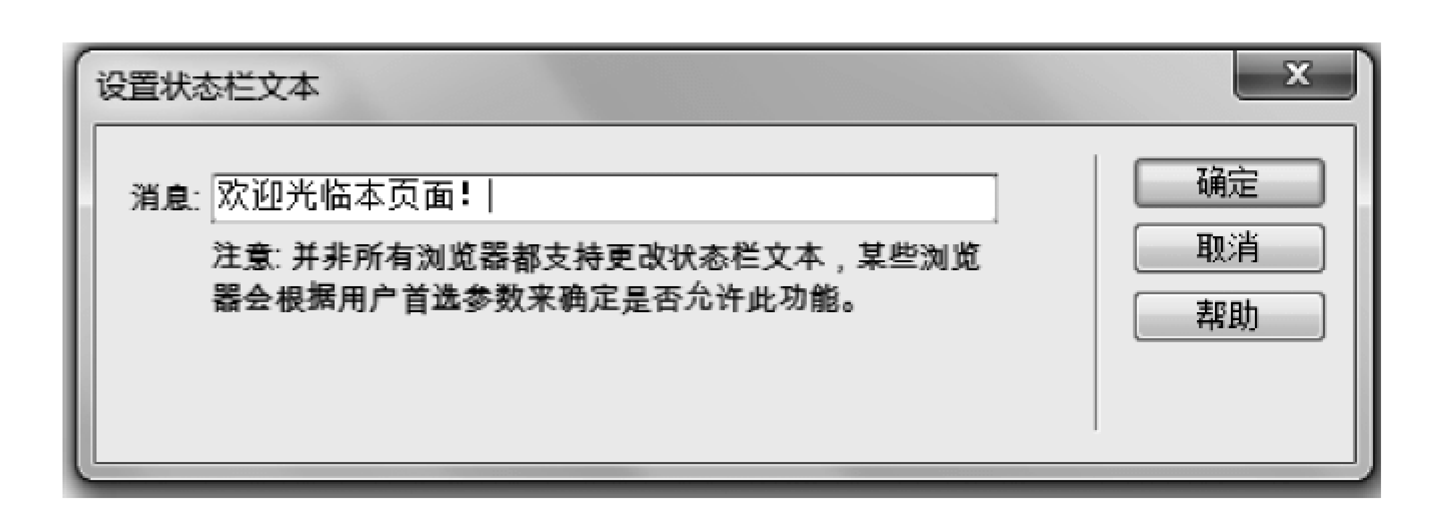


图 5.9 “设置状态栏文本”对话框

(3) 单击“确定”按钮，完成设置。此时在“行为”面板中可以看到添加了一项设置状态栏文本动作，触发事件为 onLoad，如图 5.10 所示。

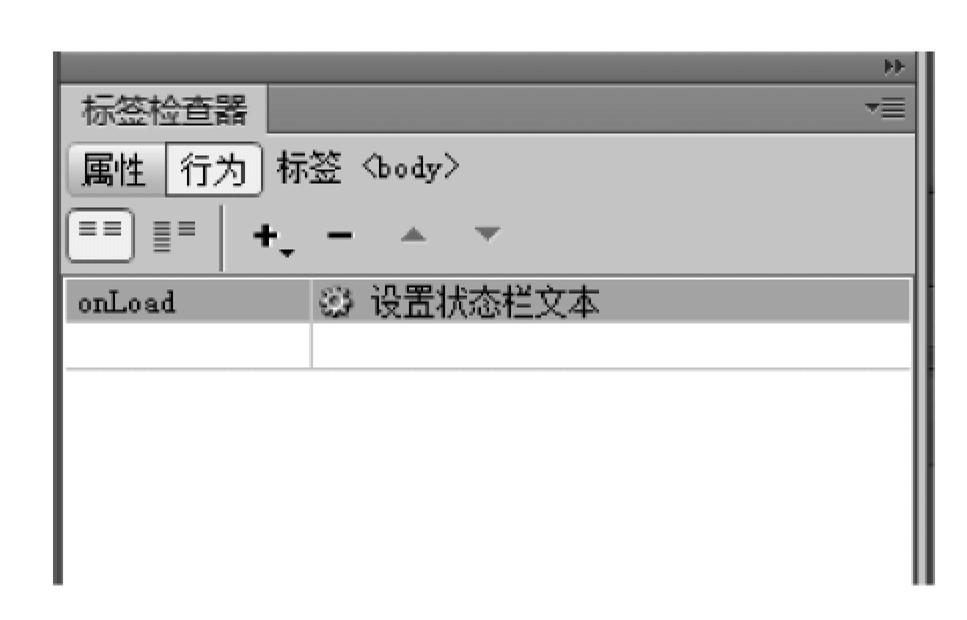


图 5.10 “行为”面板

(4) 执行“文件”|“保存”命令，保存页面。在浏览器中预览页面，可以看到在浏览器状态栏上出现了设置的状态栏文本，如图 5.11 所示。



图 5.11 查看状态栏文本效果

5.2　Spry 效果

Spry 是一个 Dreamweaver CS6 中内置的 JavaScript 库，网页设计人员可以使用它构建页面效果更加丰富的网站，例如创建菜单栏、可折叠面板等构件。在 Dreamweaver CS6 中使用 Spry 构件比较简单，但要求用户具有 HTML、CSS 和 JavaScript 的相关基础知识。

Spry 构件是网页中的一个页面元素，通过使用 Spry 构件可以轻松实现更加丰富的网页交互效果，Spry 构件主要由以下几个部分组成。

(1) 构件结构，用来定义 Spry 构件结构组成的 HTML 代码块。

(2) 构件行为，用来控制 Spry 构件如何响应用户启动事件的 JavaScript 脚本。

(3) 构件样式，用来指定 Spry 构件外观的 CSS 样式。

在 Dreamweaver CS6 的“插入”面板中的 Spry 选项卡中提供了 5 种 Spry 构件，如图 5.12 所示。



图 5.12　5 种 Spry 构件

它们分别是“Spry 菜单栏”、“Spry 选项卡式面板”、“Spry 折叠式”、“Spry 可折叠面板”、“Spry 工具提示”。

在 Dreamweaver 中插入 Spry 构件时，会自动将相关的文件链接到页面中，以便 Spry 构件中包含该页面的功能和样式。Spry 中的每个构件都与唯一的 CSS 和 JavaScript 文件相关联。在 JavaScript 脚本文件中实现了构件的相关功能，而在 CSS 样式表文件中设置了构件的外观样式。

注意：当在页面中插入 Spry 构件时，Dreamweaver CS6 会自动在站点的根目录下创建一个名称为 SpryAssets 的目录，并将相应的 CSS 样式表文件和 JavaScript 脚本文件保存在该文件夹中。

5.2.1　Spry 菜单栏

Spry 菜单栏是一组可导航的菜单按钮，可以使页面在有限的空间内显示大量的导航信息，当鼠标指向某个按钮时，即可弹出子菜单的项目，如图 5.13 所示。

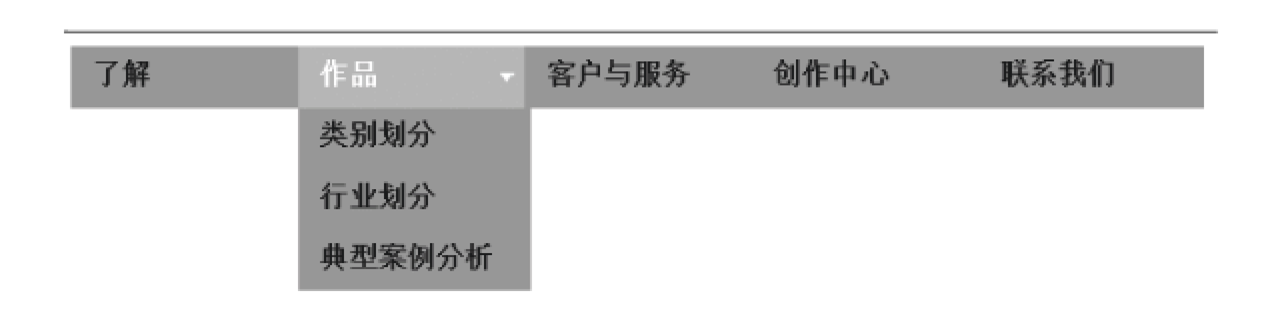


图 5.13　菜单栏效果

1. 插入 Spry 菜单栏

使用 Spry 菜单栏可以在紧凑的空间中显示大量的导航信息，并且使浏览者能够清楚网站中的站点目录结构。当用户将鼠标移至某个菜单栏按钮时，将显示相应的子菜单。

步骤如下：

(1) 将光标放置在页面中需要插入 Spry 菜单栏的位置，单击“插入”面板上的 Spry 选项卡中的“Spry 菜单栏”按钮，如图 5.14 所示。弹出“Spry 菜单栏”对话框，如图 5.15 所示。



图 5.14 “插入”面板



图 5.15 “Spry 菜单栏”对话框

(2) 在“Spry 菜单栏”对话框中可以选择需要插入的两种菜单栏构件，选中其中一个选项，单击“确定”按钮，即可在页面中插入 Spry 菜单栏，如图 5.16 所示。

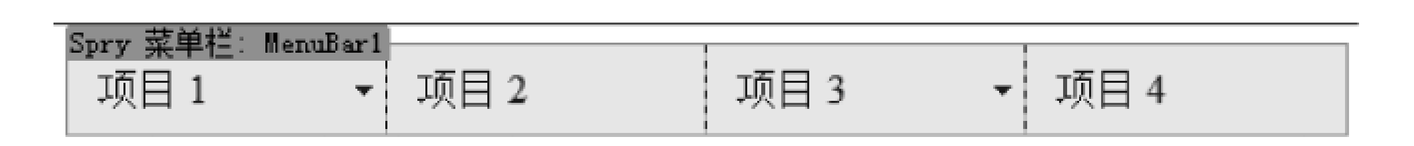


图 5.16 插入 Spry 菜单栏

2. 设置 Spry 菜单栏属性

选中刚刚插入的 Spry 菜单栏，在“属性”面板上可以对其相关属性进行设置，如图 5.17 所示。

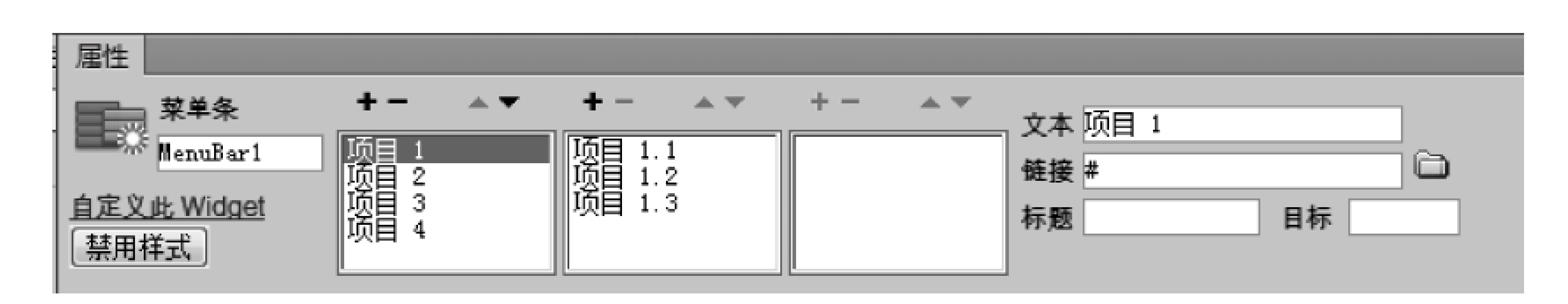


图 5.17 Spry 菜单栏的“属性”面板

菜单条：在该文本框中可以为 Spry 菜单栏命名。默认情况下，插入到页面中的菜单栏会以 MenuBar1、MenuBar2 等的命名规则进行命名。

禁用样式：单击该按钮即可禁用 Spry 菜单栏的 CSS 样式，以便可以在 Dreamweaver CS6 的设计视图中查看菜单栏的 HTML 结构。注意，单击“禁用样式”按钮，该按钮将变为“启用样式”按钮。

菜单栏列表：在“属性”面板的中间位置有三个列表，从左至右分别为“主菜单栏列表”、“子菜单栏列表”、“三级菜单栏列表”，在每个菜单项列表中可以对相应的菜单项进行添加、

删除以及调整顺序等操作。

文本：在该文本框中可以更改选中的菜单项的名称。

链接：在该文本框中可以为选中的菜单项设置其相应的链接，或者单击“浏览”按钮，在弹出的“选择文件”对话框中选择需要的链接文件。

标题：在该文本框中可以为选中的菜单项设置提示文本。

目标：在该文本框中可以输入链接打开的方式。例如，可以为菜单项分配一个目标属性，以便浏览者单击链接时，在新的浏览器窗口中打开链接的页面。

3. 调整菜单项样式

在 Dreamweaver CS6 中提供的 Spry 构件，默认的样式如果不能满足用户的需求，可以通过对 Spry 构件所生成的 CSS 样式表和该 Spry 构件的显示外观进行适当调整和修改。

在设计窗口中选中需要禁用样式的菜单栏构件，单击“属性”面板上的“禁用样式”按钮，可以采用禁用菜单样式，当禁用菜单样式后，菜单项则会以项目符号的列表形式显示在页面上，而不是显示为菜单栏中带样式的菜单项。效果如图 5.18 所示。

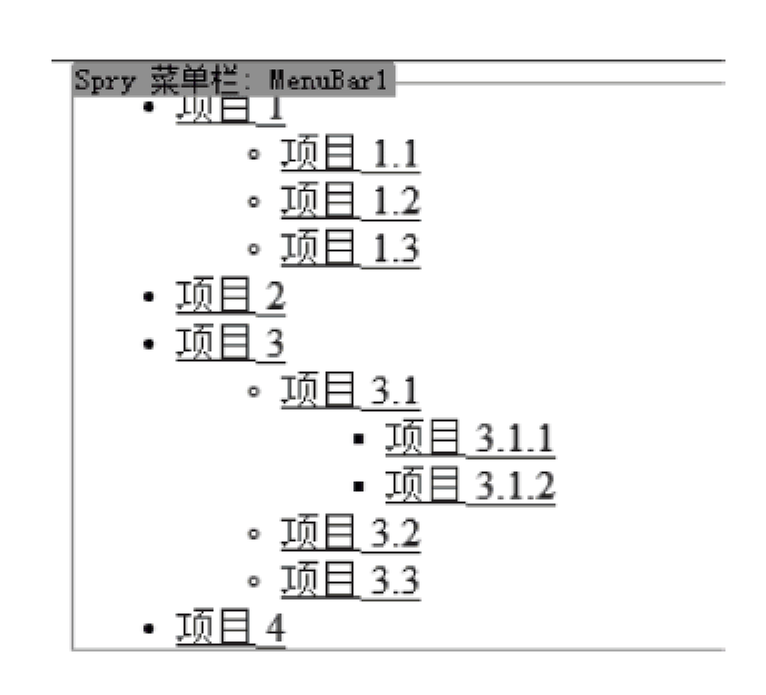


图 5.18 Spry 菜单效果

按钮转变为“启用样式”按钮，选中要启用样式的菜单栏构件，单击“属性”面板上的“启用样式”按钮，可以重新启用默认的 Spry 菜单栏样式。

为<a>标签应用的 CSS 样式中包含 Spry 菜单项文本的相关样式，要更改 Spry 菜单项的文本样式，可以通过修改 CSS 样式更改其默认值。Spry 菜单栏文本的 CSS 样式见表 5.1。

表 5.1 Spry 菜单栏的 CSS 样式(1)

要更改的 CSS 样式	垂直或水平 Spry 菜单栏的 CSS 规则	相关属性和默认值
默认文本	ul. MenuBarVertical a ul. MenuBarHorizontal a	color：#333； text-decoration：none；
当鼠标移过文本时，文本的颜色	ul. MenuBarVertical a:hover ul. MenuBarHorizontal a:hover	color：#FFF；
具有焦点的文本颜色	ul. MenuBarVertical a:focus ul. MenuBarHorizontal a:focus	color：#FFF；
当鼠标指针移到菜单项时，菜单栏的颜色	ul. MenuBarVertical a. MenuBarItemHover ul. MenuBarHorizontal a. MenuBarItemHover	color：#FFF；
当鼠标指针移过子菜单项时，子菜单项的文本颜色	ul. MenuBarVertical a. MenuBarItemSubmenuhover ul. MenuBarHorizontal a. MenuBarItemSubmenuhover	color：#FFF；

为<a>标签应用的 CSS 样式中包含 Spry 菜单项背景颜色的相关样式，要更改 Spry 菜单项的背景颜色，可以通过表 5.2 来查找相应的 CSS 样式，然后更改其默认值。Spry 菜单栏背景的 CSS 样式见表 5.2。

表 5.2 Spry 菜单栏的 CSS 样式(2)

要更改的背景	垂直或水平 Spry 菜单栏的 CSS 规则	相关属性和默认值
默认背景	ul. MenuBarVertical a ul. MenuBarHorizontal a	background-color: #EEE;
当鼠标移过文本时,背景的颜色	ul. MenuBarVertical a:hover ul. MenuBarHorizontal a:hover	background-color: #33C;
具有焦点的菜单项的背景颜色	ul. MenuBarVertical a:focus ul. MenuBarHorizontal a:focus	background-color: #33C;
当鼠标指针移到菜单项时,菜单栏项的背景颜色	ul. MenuBarVertical a. MenuBarItemHover ul. MenuBarHorizontal a. MenuBarItemHover	background-color: #33C;
当鼠标指针移过子菜单项时,子菜单项的背景颜色	ul. MenuBarVertical a. MenuBarItemSubmenuhover ul. MenuBarHorizontal a. MenuBarItemSubmenuhover	background-color: #33C;

4. 制作网页下拉菜单实例

(1) 执行"文件"|"新建"命令,新建一个空白的 HTML 页面,将该文件保存。单击"插入"面板上的 Spry 选项卡中的"Spry 菜单栏"按钮,在页面中插入 Spry 菜单栏,如图 5.19 所示。

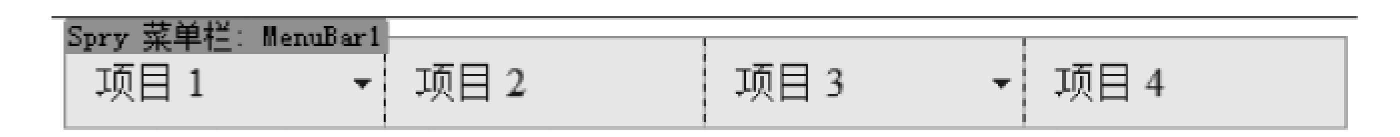


图 5.19 插入 Spry 菜单栏

(2) 选中刚刚插入的 Spry 菜单栏,在"属性"面板上的"主菜单栏列表"框中选择"项目 1"选项,可以在"子菜单栏列表"框中看到该菜单下的子菜单项,如图 5.20 所示。

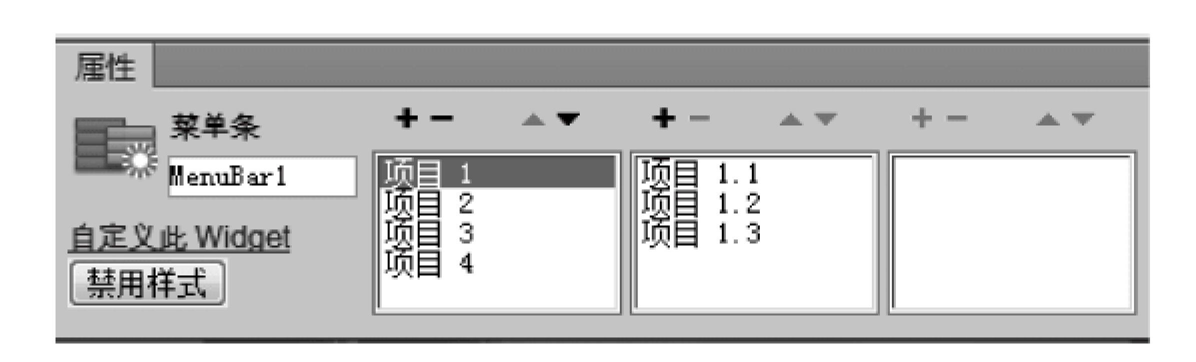


图 5.20 显示子菜单项

在"子菜单栏列表"框中选中需要删除的项目,单击其上方的"删除菜单项"按钮 - ,可以删除选中的子菜单项,如图 5.21 所示。

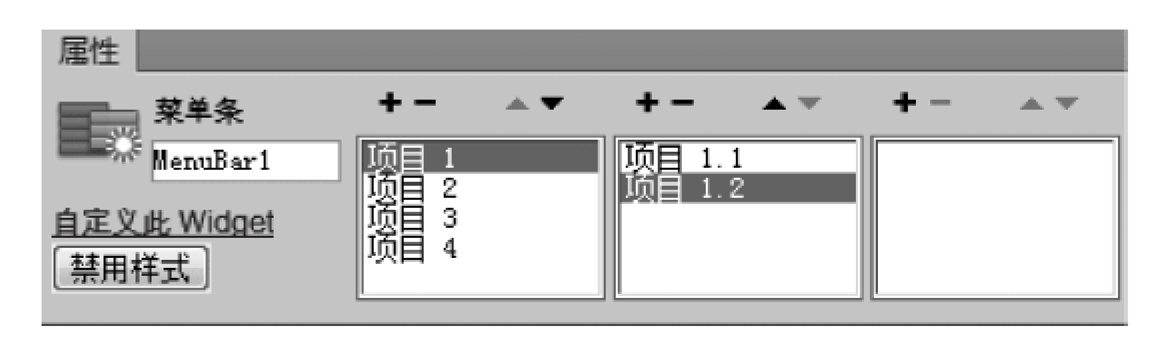


图 5.21 删除子菜单项

(3) 在"主菜单栏列表"框中选择"项目 1"选项,在"文本"文本框中修改该菜单项的名称。使用相同的制作方法,修改其他各主菜单项的名称,如图 5.22 所示。

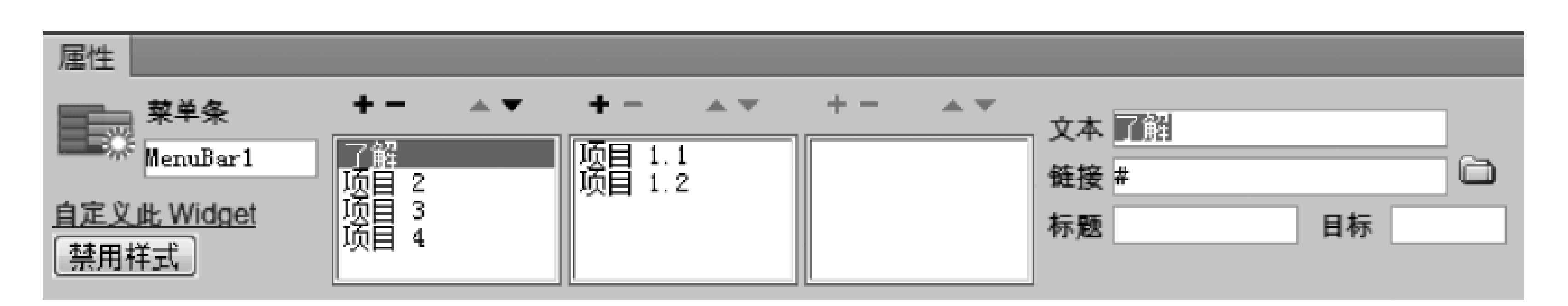


图 5.22 修改菜单项名称

（4）单击“主菜单栏列表”框上的“添加菜单项”按钮 **+**，可以添加相应的主菜单项，如图 5.23 所示。



图 5.23 添加主菜单项

在“主菜单栏列表”框中选中某个主菜单项，在“子菜单栏列表”框中可以添加相应的子菜单项，如图 5.24 所示。



图 5.24 添加子菜单项

（5）使用相同的制作方法，完成 Spry 菜单栏中各菜单项的设置，如图 5.25 所示。切换到 Spry 菜单栏的外部 CSS 样式表文件 SpryMenuBarHorizontal. css 中，找到 ul. MenuBarHorizontal a. MenuBar ItemSubmenu 和 ul. MenuBarHorizontal ul 样式表，如图 5.26 所示，将这两个 CSS 样式设置删除。



图 5.25 菜单栏的效果

```
ul.MenuBarHorizontal a.MenuBarItemSubmenu
{
    background-image: url(SpryMenuBarDown.gif);
    background-repeat: no-repeat;
    background-position: 95% 50%;
}
```

```
ul.MenuBarHorizontal ul
{
    border: 1px solid #CCC;
}
```

图 5.26 CSS 样式代码

(6) 再找到 ul. MenuBarHorizontal a 样式表,如图 5.27 所示。对样式进行相应修改,修改后如图 5.28 所示。

```
ul.MenuBarHorizontal a
{
    display: block;
    cursor: pointer;
    background-color: #EEE;
    padding: 0.5em 0.75em;
    color: #333;
    text-decoration: none;
}
```

图 5.27 CSS 样式代码

```
ul.MenuBarHorizontal a
{
    color:#fff;
    font-weight:bold;
    display: block;
    cursor: pointer;
    background-color: #9c0;
    border:block;
    padding: 0.5em 0.75em;
    color: #333;
    text-decoration: none;
}
```

图 5.28 修改后的 CSS 样式代码

(7) 返回到设计视图,可以看到下拉菜单的效果。切换到外部 CSS 样式表文件 SpryMenuBarHorizontal.css,找到相应的样式表,如图 5.29 所示。

```
ul.MenuBarHorizontal a.MenuBarItemHover,
ul.MenuBarHorizontal a.MenuBarItemSubmenuHover,
ul.MenuBarHorizontal a.MenuBarSubmenuVisible
{
    background-color: #33C;
    color: #FFF;
}
```

图 5.29 CSS 样式代码

(8) 对样式进行相应的修改,修改后如图 5.30 所示。返回设计视图,执行"文件"|"保存"命令,弹出"复制相关文件"对话框,如图 5.31 所示。

```
ul.MenuBarHorizontal a.MenuBarItemHover,
ul.MenuBarHorizontal a.MenuBarItemSubmenuHover,
 ul.MenuBarHorizontal a.MenuBarSubmenuVisible
{
    background-color: #ccc;
    color: #FFF;
}
```

图 5.30 修改后的 CSS 样式代码



图 5.31 "复制相关文件"对话框

(9) 单击“确定”按钮，在浏览器中预览页面，可以看到 Spry 菜单栏的效果，如图 5.32 所示。

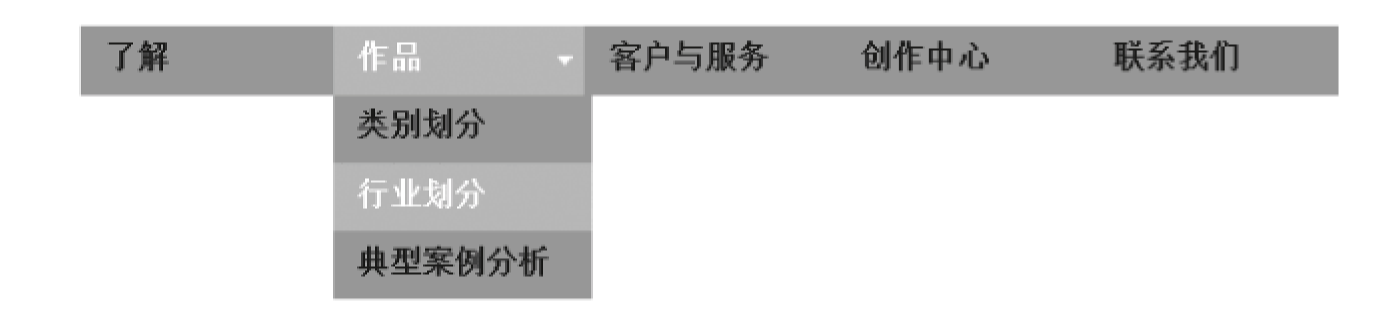


图 5.32　预览效果

5.2.2　Spry 选项卡式面板

Spry 选项卡式面板构件是一种面板，用来将较多内容放置在紧凑的控件中，当浏览者单击不同的选项卡标签时，即可打开相应的面板。浏览者可以通过单击面板选项卡标签隐藏或显示放置在选项卡式面板中的内容。

1. 插入选项卡式面板

将光标放置在页面中需要插入 Spry 选项卡式面板的位置，单击“插入”面板上的 Spry 选项卡中的“Spry 选项卡式面板”按钮，如图 5.33 所示。在页面中插入 Spry 选项卡式面板，如图 5.34 所示。

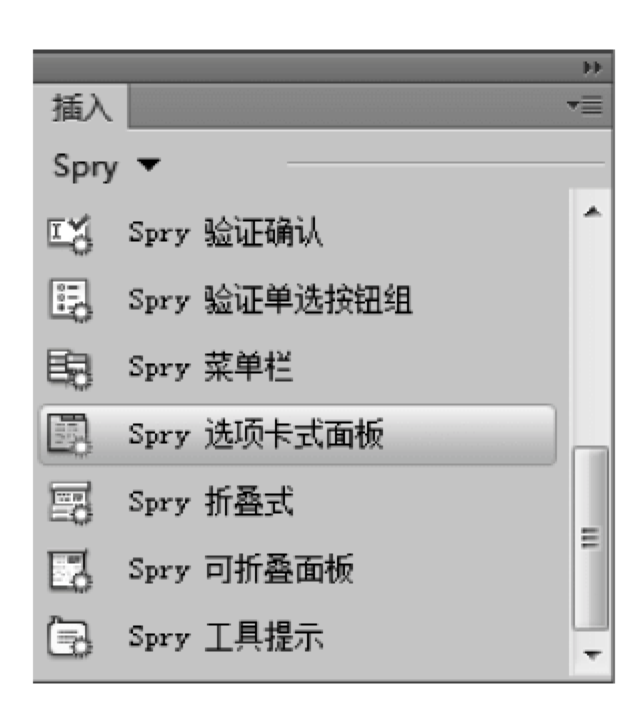


图 5.33　“插入”面板

Spry 选项卡式面板: TabbedPanels1
标签 1　标签 2
内容 1

图 5.34　插入 Spry 选项卡式面板

2. 设置选项卡式面板的属性

选中页面中刚插入的 Spry 选项卡式面板，在“属性”面板上可以对 Spry 选项卡式面板的相关属性进行设置，如图 5.35 所示。

选项卡式面板：在该文本框中可以为 Spry 选项卡式面板命名。默认情况下，插入页面中的选项卡式面板会以 TabbedPanels1、TabbedPanels2 等的命名规则命名。



图 5.35　Spry 选项卡式面板的“属性”面板

面板：在该列表中显示了该 Spry 选项卡式面板的各面板，单击其上方的“添加面板”按钮 + ，即可添加面板；在列表中选中某个面板，单击列表上方的“删除面板”按钮 − ，即可删除该面板，另外，还可以调整面板的前后顺序。

默认面板：在该选项的下拉菜单中列出了 Spry 选项卡式面板的所有面板名称，选择某个面板名称，在浏览器中预览页面时，在默认的情况下，所设置的面板将会显示，其他面板则被隐藏。

执行“文件”|“保存”命令，保存页面，弹出“复制相关文件”对话框，在该对话框中列出了 Spry 选项卡式面板中所用到的 JavaScript 脚本文件和外部 CSS 样式表文件，单击“确定”按钮，在浏览器中预览页面，可以看到 Spry 选项卡式面板的效果，如图 5.36 所示。



图 5.36　预览 Spry 选项卡式面板

可以通过修改相关的 CSS 样式来更改 Spry 选项卡式面板的外观样式。需要切换到 Spry 选项卡式面板的外部样式表文件 SpryTabbedPanels. css 中，其中，. TabbedPanelsTab 样式表主要定义了选项卡式面板标签的默认状态；. TabbedPanelsTabSelected 样式表主要定义了选项卡式面板中当前选中标签的状态；. TabbedPanelsContentGroup 样式表定义了选项卡式面板内容部分的外观。

5.2.3　Spry 折叠式

Spry 折叠式面板可以将大量页面内容放置在一个紧凑的空间中，从而达到节省空间的目的。浏览者只需单击该构件的选项卡标签，就可以显示或隐藏该面板的内容。当浏览者单击不同的选项卡标签时，折叠式构件的面板会相应展开或收缩。

1. 插入 Spry 折叠式构件

将光标放置在页面中需要插入 Spry 折叠式构件的位置，单击“插入”面板上的 Spry 选项卡中的“Spry 折叠式”按钮，即可在页面中插入 Spry 折叠式构件，如图 5.37 所示。

2. 设置 Spry 折叠式构件属性

选中页面中刚插入的 Spry 折叠式构件，在“属性”面板上可以对其相关属性进行设置，如图 5.38 所示。

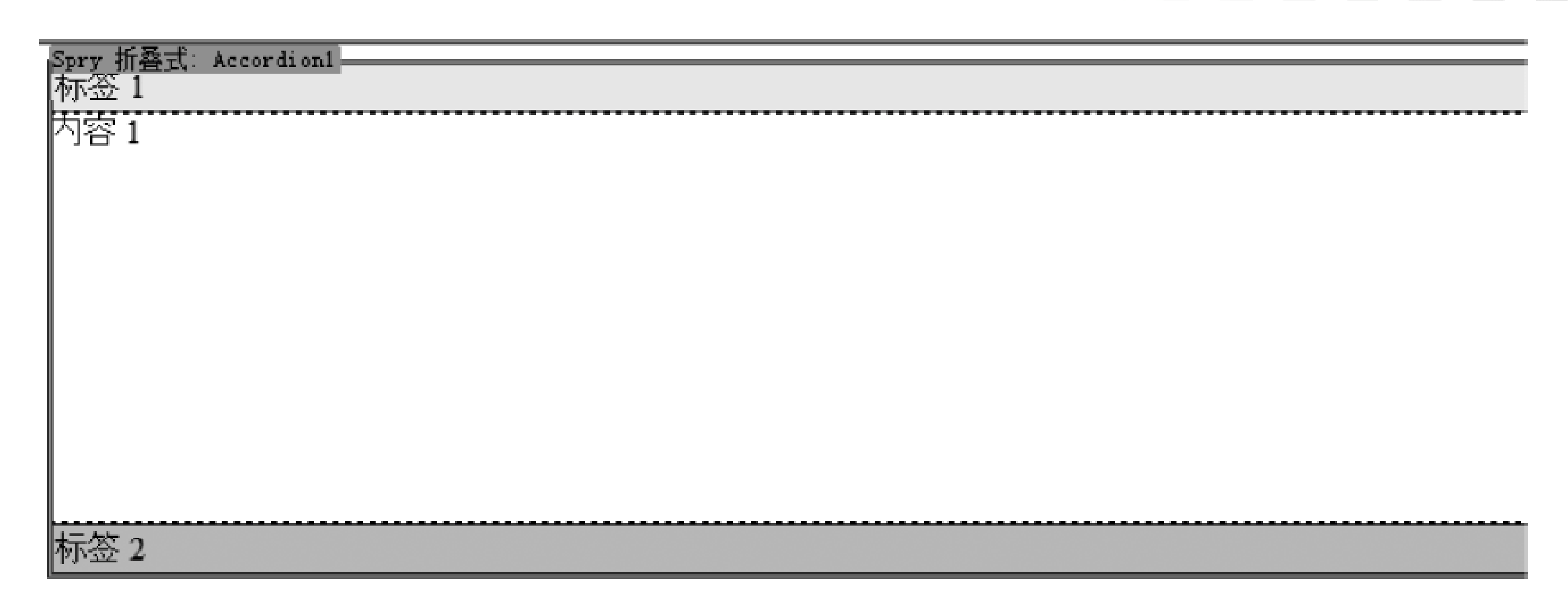


图 5.37　插入 Spry 折叠式构件

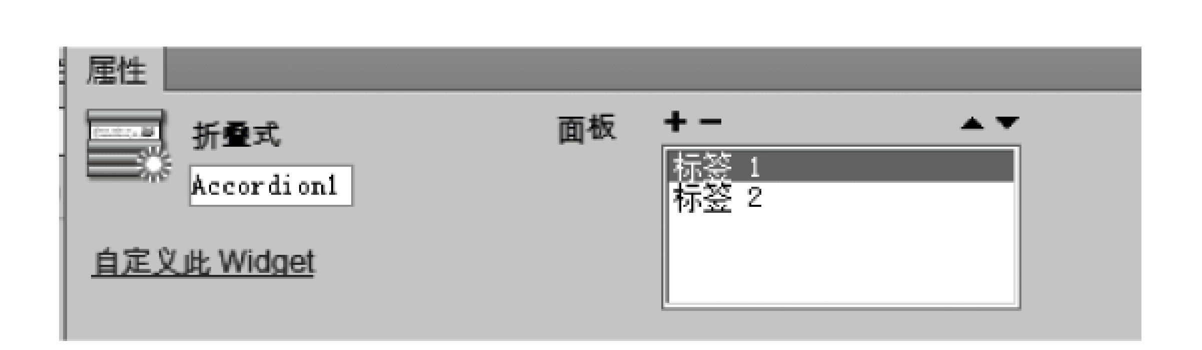


图 5.38　Spry 折叠式"属性"面板

折叠式：在该文本框中可以为 Spry 折叠式构件进行命名。默认情况下，插入到页面中的折叠式构件会以 Accordion1、Accordion2 的命名规则进行命名。

面板：在该选项的列表中列出了所选中的 Spry 折叠式构件中的各面板，单击其上方的"添加面板"按钮 + ，即可添加面板；在列表中选中某个面板，单击列表上方的"删除面板"按钮 − ，即可删除该面板。另外，还可以调整面板的前后顺序。

执行"文件"|"保存"命令，保存页面，弹出"复制相关文件"对话框，在该对话框中列出了 Spry 选项卡式面板中所用到的 JavaScript 脚本文件和外部 CSS 样式表文件，单击"确定"按钮，在浏览器中预览页面，可以看到 Spry 折叠式构件的效果，如图 5.39 所示。



图 5.39　Spry 折叠式面板的预览效果

3. 设置 Spry 折叠式样式

如果需要对 Spry 折叠式构件的文本样式进行设置和调整，可以通过设置整个折叠式构件容器的属性，或分别设置构件的各组件的属性来实现。

要更改 Spry 折叠式构件的文本样式，可以修改相关的 CSS 样式，添加文本样式属性和值即可。需要切换到 Spry 选项卡式面板的外部样式表文件 SpryAccordion.css 中，其中，Accordion 样式表主要定义了整个折叠构件(包括选项卡和内容面板)中的文本的默认状态；.AccordionPanelTab 样式表主要定义了折叠式面板选项卡中的文本的状态；.AccordionPanelContent 样式表定义了折叠式内容面板中的文本的状态。

5.2.4 Spry 可折叠面板

Spry 可折叠面板与 Spry 可折叠式相似，都是将页面放在一个小的空间里，达到节省页面空间的作用，只是在外观上有所区别。

1. 插入 Spry 可折叠面板

将光标放置在页面中需要插入 Spry 可折叠面板的位置，单击“插入”面板上的 Spry 选项卡中的“Spry 可折叠面板”按钮，即可在页面中插入 Spry 可折叠面板，如图 5.40 所示。

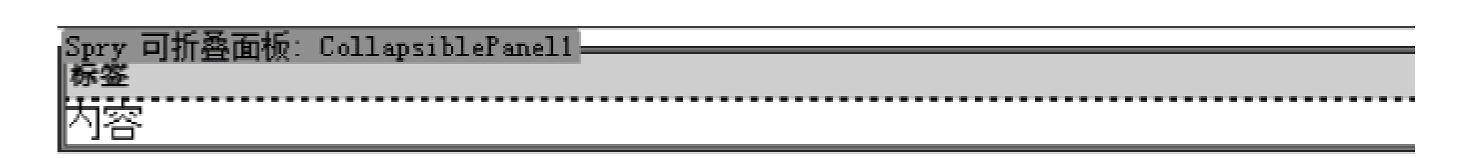


图 5.40 插入 Spry 可折叠面板

2. 设置 Spry 可折叠面板属性

选中页面中刚插入的 Spry 可折叠面板，在“属性”面板上可以对其相关属性进行设置，如图 5.41 所示。

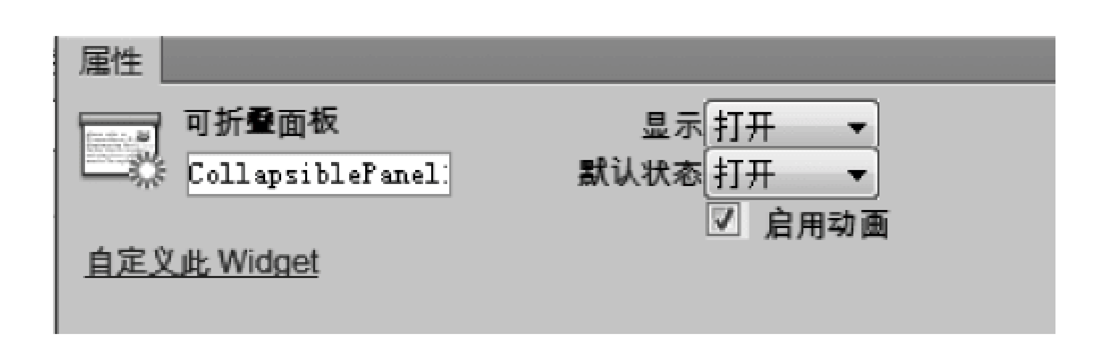


图 5.41 Spry 可折叠面板属性

可折叠面板：在该文本框中可以为 Spry 可折叠面板进行命名。默认情况下，插入到页面中的可折叠面板会以 CollapsiblePanel1、CollapsiblePanel2 的命名规则进行命名。

显示：该选项可以设置 Spry 可折叠面板中在设计视图中是打开的还是关闭的。在该选项的下拉列表中有两个选项，分别是“打开”和“已关闭”，默认情况下，选择“打开”选项，如果在下拉列表中选择“已关闭”选项，则该 Spry 可折叠面板在设计视图中是关闭的，如图 5.42 所示。

Spry 可折叠面板: CollapsiblePanel1
标签

图 5.42 “显示”属性为“已关闭”

默认状态：该选项主要用于设置在浏览器中浏览该 Spry 可折叠面板时，可折叠面板的默认状态，在该选项的下拉列表中同样有两个选项，分别是“打开”和“已关闭”，默认情况下，选择“打开”选项。

启用动画：选中该复选框，浏览者在单击该面板选项卡标签时，该面板将缓慢地平滑打开和关闭；如果没有选中该复选框，则浏览者在单击该面板选项卡标签时，可折叠面板会迅速打开和关闭，默认情况下，为选中该复选框。

执行“文件”|“保存”命令，保存页面，弹出“复制相关文件”对话框，在该对话框中列出了 Spry 可折叠面板中所用到的 JavaScript 脚本文件和外部 CSS 样式表文件，单击“确定”按钮，在浏览器中预览页面，可以看到 Spry 可折叠面板的效果，如图 5.43 和图 5.44 所示。



图 5.43 Spry 可折叠面板的预览效果(1)



图 5.44 Spry 可折叠面板的预览效果(2)

3. 设置 Spry 可折叠面板的样式

如果想设置 Spry 可折叠面板的文本样式，可以通过设置整个 Spry 可折叠面板构件容器的属性或分别设置构件的各个组件的属性来实现。

如果需要更改可折叠面板的文本样式，可以修改相关的 CSS 样式，添加文本样式属性和值即可。需要切换到 Spry 选项卡式面板的外部样式表文件 SpryCollapsiblePanel.css

中,其中.CollapsiblePane样式表定义了整个可折叠面板中的文本的默认状态;.CollapsiblePaneTab样式表定义了面板选项卡中的文本的状态;.CollapsiblePaneContent样式表定义了内容面板中的文本的状态。

5.3 JavaScript 介绍

JavaScript是一种可以嵌入在HTML代码中由客户端浏览运行的脚本语言。之前介绍的丰富多彩的行为效果,其实都是Dreamweaver在源代码中加入JavaScript实现的。JavaScript是一种基于对象和事件驱动并且有安全性能的脚本语言。使用它的目的是与HTML一起实现在一个Web页面与Web客户交互。它是通过嵌入在标准的HTML中实现的。它的出现弥补了HTML的缺陷。JavaScript是一种比较简单的编程语言。使用方法是向Web页面的HTML文件中增加一个脚本,当一个支持JavaScript的浏览器打开这个页面时,它会读出这个脚本并执行其命令。

在网页中使用JavaScript代码,不仅可以实现网页特效,还可以响应用户请求实现动态交互的功能。在PHP动态网页中灵活运用JavaScript,可以实现更强大的功能。本节将介绍JavaScript脚本语言的基本知识,使读者在掌握基础内容的前提下能够简单运用JavaScript制作Web页面。

5.3.1 JavaScript 概述

1. 什么是 JavaScript

JavaScript是由NetScape Communication Corporation(网景公司)开发的,是一种基于对象和事件驱动并具有安全性能的解释型脚本语言。它不但可用于编写客户端的脚本程序,由Web浏览器解释执行,而且还可以编写在服务器端执行的脚本程序,在服务器端处理用户提交的信息并动态地向浏览器返回处理结果。

2. JavaScript 的功能

JavaScript是比较流行的一种制作网页特效的脚本语言,它由客户端浏览器解释执行,可以应用在PHP、ASP、JSP和ASP.NET网站中,同时目前比较热门的Ajax就是以JavaScript为基础,由此可见,熟练掌握并应用JavaScript对于网站开发人员非常重要。

JavaScript主要应用于以下几个方面。

(1) 在网页中加入JavaScript脚本代码,可以使网页具有动态交互的功能,便于网站与用户间的沟通,及时响应用户的操作,对提交的表单做及时的检查,如验证表单元素是否为空,验证表单元素是否是数值型、检测表单元素是否输入错误等。

(2) 应用JavaScript脚本制作网页特效,如动态的菜单、浮动的广告等,为页面增添绚丽的动态效果,使网页内容更加丰富、活泼。

(3) 应用JavaScript脚本建立复杂的网页内容,如打开新窗口载入网页。

(4) 应用JavaScript脚本可以对用户的不同事件产生不同的响应。

(5) 应用JavaScript制作各种各样的图片、文字、鼠标、动画和页面的效果。

（6）应用 JavaScript 可以制作一些小游戏。

5.3.2 JavaScript 编程基础

JavaScript 的语法非常简单，所有 JavaScript 都是在 HTML 代码中编写的，使用<script>标记完成。

【例 5.1】 第一个 JavaScript 程序——script_5_01.html。

```
<html>
  <head>
      <title> JavaScript 程序 1 </title>
       <script language = "JavaScript">
           alert("Welcome! ");
       </script>
</head>
</html>
```

其中，<script>表示脚本语言的开始，使用 language 属性定义脚本语言为 JavaScript，在标记<script language="JavaScript">与</script>之间就可以加入 JavaScript 脚本，一般而言，<script>标记都是出现在<head>标记中的，也可以在任意位置上编写，但是最好在调用其操作之前进行编写。

本实例设置弹出信息，alert ( )是 JavaScript 的窗口对象方法，其功能是弹出一个具有“确定”按钮的对话框并显示括号中的字符串，预览效果如图 5.45 所示。



图 5.45 弹出对话框

如果一个 HTML 页面中编写包含很多的 JavaScript 代码，可以考虑将一些 JavaScript 代码单独定义成 *.js 文件，然后在需要的页面中导入即可。

【例 5.2】 定义 *.js 文件——welcome.js。

```
alert("Welcome! ");                              //弹出一个对话框
```

【例 5.3】 在<script>标记中使用 src 属性导入需要的 *.js 文件——script_5_02.html。

```
<html>
  <head>
      <title> JavaScript 程序 2 </title>
      <script language = "JavaScript" src = "welcome.js">
      </script>
  </head>
</html>
```

运行本程序时，将 welcome.js 中定义的 JavaScript 代码导入到页面中执行，运行效果和例 5.1 效果相同。

5.3.3 调用 JavaScript 脚本

在 JavaScript 开发中最常用的部分是函数，也是代码中最常用的一种形式。函数的定义语法格式如下：

```
function 函数名称(参数 1,参数 2,…){
  [return 返回值];
}
```

在 JavaScript 中定义的函数不需要声明返回值类型,如果一个函数需要有返回值,则通过 return 语句返回即可。

【例 5.4】 定义两个数字相加的函数——script_5_03.html。

```
<html>
  <head>
      <title>JavaScript 程序 3</title>
      <script language="JavaScript">
            function add(i,j){
               return i+j;
            }
            alert("数字相加的结果为: "+ add(3,5));
      </script>
  </head>
</html>
```

在程序的 add()函数中接收两个参数,通过 return 返回这两个数字相加的结果,程序的运行结果如图 5.46 所示。

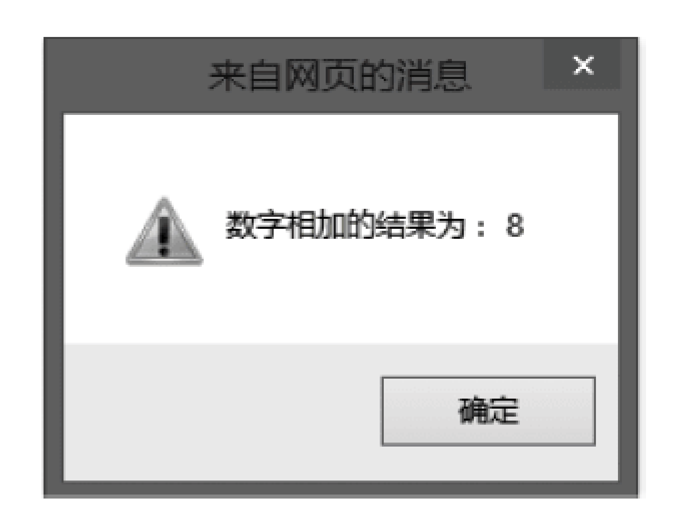


图 5.46 调用函数运行结果

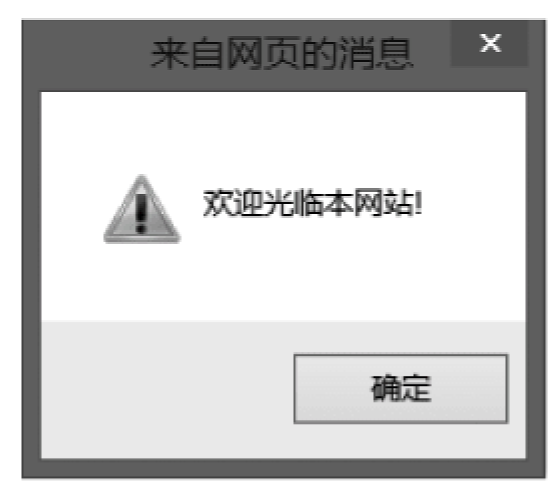
JavaScript 是一种基于对象和事件驱动并且有安全性能的脚本语言。事件可以使页面获得更好的交互效果。JavaScript 的事件处理主要围绕函数展开,一旦事件被触发后,则会根据事件的类型来调用相应的函数来完成事件的处理操作。

【例 5.5】 简单的事件处理程序——script_5_04.html。

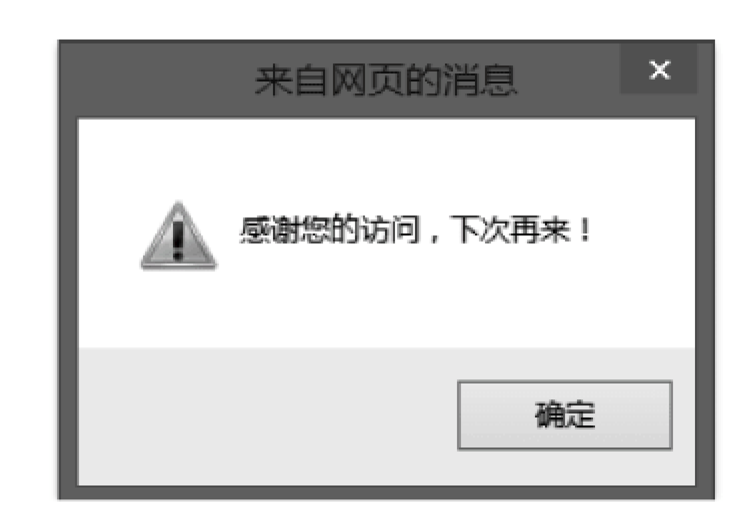
```
<html>
  <head>
      <title>JavaScript 程序 4</title>
      <script language="JavaScript">
            function hello(){
               alert("欢迎光临本网站! ");
            }
            function bye(){
               alert("感谢您的访问,下次再来!");
            }
      </script>
  </head>
  <body onLoad="hello()" onUnLoad="bye()">
  </body>
</html>
```

本程序在<body>标签中增加了两个事件,其中 onLoad 表示网页加载时要触发的事件,一旦触发事件发生后调用的是 hello()函数;onUnLoad 表示关闭页面时要触发的事件,

一旦触发事件发生后调用的是 bye()函数。这样，当本页面打开时出现如图 5.47(a)所示的页面，当本页面关闭时出现如图 5.47(b)所示的页面。



(a) 页面打开时弹出



(b) 页面关闭时弹出

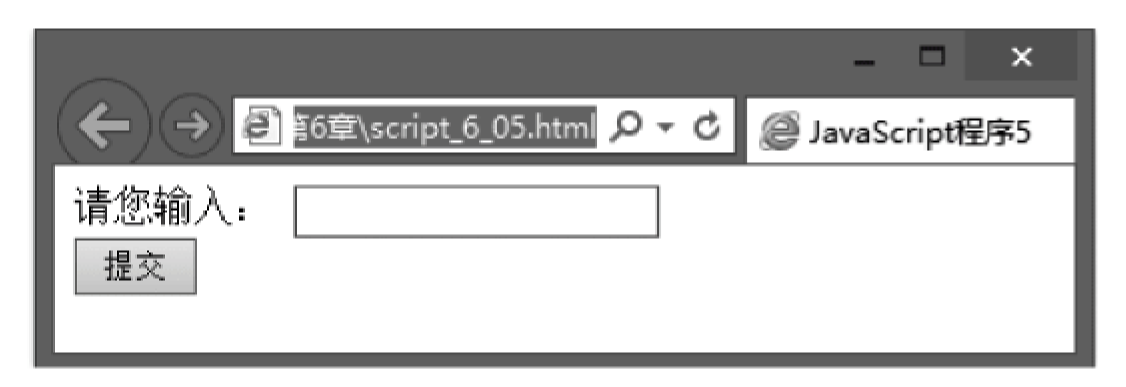
图 5.47　网页消息

在 JavaScript 中有一个 OnSubmit 事件比较常用，此事件主要用在表单提交时触发。用来检验表单提交的输入是否合法，如文本框中要求输入的数据必须为数值型数据等。

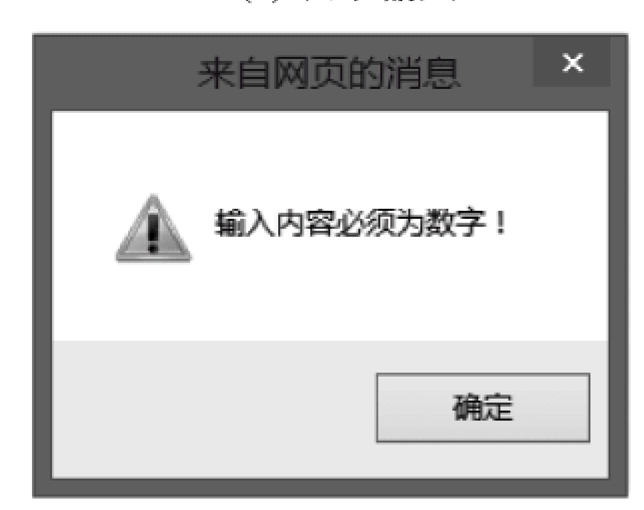
【例 5.6】　数值型数据输入提交验证——script_5_05.html。

```
<html>
  <head>
     <title> JavaScript 程序 5 </title>
     <script language = "JavaScript">
         function validate(f){                   //定义函数
            var content1 = f.content.value;      //取得输入的内容
            if(!(/^\d + $/.test(content1))){     //对输入的内容进行验证,采用正则表达式
                alert("输入内容必须为数字!");    //弹出警告框
                f.content.focus();               //让焦点定位到 content 文本框
                return false;                    //返回 false,表单不提交
            }
           return true;                          //返回 true,表单提交
     }
   </script>
</head>
<body>
    <form action = "" method = "post" onSubmit = "return validate(this)">
请您输入数字型数据:
    <input type = "text" name = "content"><br>
    <input type = "submit" value = "提交">
</form>
</body>
</html>
```

本程序的<form>元素中使用了 OnSubmit 事件在表单提交前进行验证，即该事件是<form>元素调用的，此事件决定表单是否提交，使用 return 来接收 validate()函数的返回值。如果此函数返回 true，则表示提交的数据合法，可以提交；如果此函数返回 false，则表单将不会提交。程序的运行结果如图 5.48(a)所示，如果输入非数值型数据，则会弹出警告框，效果如图 5.48(b)所示。



(a) 表单输入



(b) 弹出警告框

图 5.48 程序的运行结果

小结

本章重点介绍了 Dreamweaver 的高级应用技巧，包括行为的使用以及 Spry 效果，网页设计人员可以使用它们构建页面效果更加丰富的网站，例如，向网站页面中添加动态效果、创建菜单栏、可折叠面板等构件。无论是行为还是 Spry 效果，都需要 JavaScript 语言的支持，Spry 框架中的每一个构件都与唯一的 CSS 和 JavaScript 文件相关联，JavaScript 文件赋予构件功能。本章最后一节介绍了 JavaScript 知识，读者可对 JavaScript 做简单的了解，JavaScript 是基于对象和事件驱动的解释型脚本语言，它不但可用于编写客户端的脚本程序，也可编写在服务器端执行的脚本程序，在服务器端处理用户提交的信息并动态地向浏览器返回处理结果。

习题

1. 什么是行为？什么是事件？什么是动作？
2. 在网页文档中添加某种行为，一般需要哪些操作步骤？
3. 什么是 Spry 效果？如何在网页文档中应用 Spry 效果？
4. 什么是 JavaScript？
5. 如何定义 JavaScript 中的函数？
6. 如何对表单进行 JavaScript 验证？

第6章 数据库基础

本章要点：

- 数据库的基本概念；
- 数据模型；
- 实体联系模型；
- 关系模型。

MySQL 是关系型数据库管理系统，所使用的 SQL 是用于访问数据库的最常用标准化语言，为了更好地学习 MySQL，首先需要了解数据库的一些基本知识。

6.1 数据库的基本概念

1. 数据

数据(Data)是数据库系统研究和处理的对象，是信息的符号表示，或称为载体。数据不仅包含数字，还包括文本、图像、音频、视频等多媒体形式。

2. 数据库

数据库(DataBase，DB)是长期储存在计算机内的、有组织的(按一定的数据结构组织起来的)、可共享的数据集合。

3. 数据库管理系统

数据库管理系统(DataBase Management，DBMS)是位于用户与操作系统之间的一层数据管理软件，它为用户或应用程序提供访问数据库的方法，包括数据库的建立、查询、更新及各种数据控制。

DBMS 总是基于某种数据模型，可以分为层次型、网状型、关系型和面向对象型等。目前市场上常见的数据库管理系统包括 Oracle、DB2、Sybase、MySQL 和 SQL Server 等。

4. 数据库系统

数据库系统(DataBase System，DBS)是采用数据库技术的计算机应用系统。数据、数据库、数据库管理系统与操作数据库的应用程序，加上支撑它们的硬件平台、软件平台及与数据库有关人员，构成了一个完整的数据库系统。

6.2 数据模型

模型是对现实世界的抽象,例如汽车模型、建筑模型,它使人们能从中了解真实生活中的事物。数据模型是对数据特征的抽象,描述数据的共性。使用计算机来解决现实世界中的问题,需要把现实世界中的具体事物抽象、组织为某一数据库管理系统支持的数据模型,抽象层次如图6.1所示。

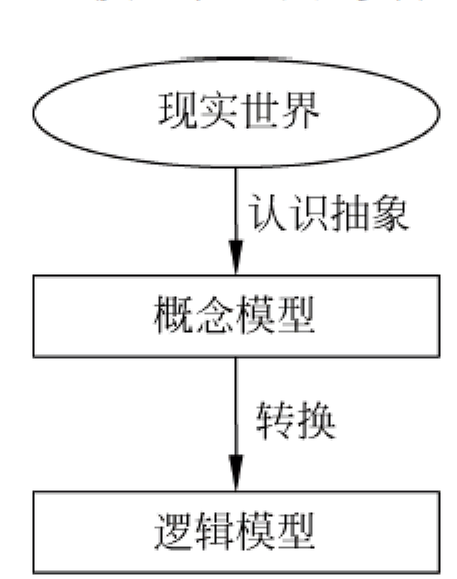


图6.1 现实世界客观对象抽象过程

1. 数据模型中的基本概念

在进行数据模型的研究中经常用到实体、属性、实体间的联系等概念,下面分别介绍。

1) 实体

实体是客观存在且可以相互区别的事物。实体可以是具体的对象,如一名学生、一辆汽车,也可以是抽象的对象,如一项工程、一次会议等。

2) 属性

实体有很多特性,每一个特性称为属性。每个属性有一个值域,其类型可以是整数型、实数型、字符类型等。例如,学生有学号、姓名、年龄等属性。

3) 实体集

性质相同的同类实体的集合称为实体集。例如,全体学生就是一个实体集。

4) 实体型

用实体名及其属性名集合来抽象和刻画同类实体,称为实体型。例如,学生(学号,姓名,性别,出生年月,所在院系)。

5) 实体标识符

能唯一标识实体的属性和属性集的标识符称为实体标识符,有时也称为关键码,或简称为键。例如,学生的学号可以作为实体的标识符。

6) 联系

联系用于描述实体之间的相互关系。与一个联系有关的实体集的个数,称为联系的元素。联系有一元联系、二元联系、三元联系。下面主要介绍一下二元联系。

二元联系有以下三种类型。

(1) 一对一联系(1∶1)

如果对于实体集 A 中的每一个实体,实体集 B 中至多有一个(也可以没有)实体与之联系,反之亦然,则称实体集 A 与实体集 B 具有一对一联系,记为1∶1。

例如,学校里面,一个班级只有一个正班长,而一个班长只在一个班中任职,则班级与班长之间具有一对一联系。

(2) 一对多联系(1∶n)

如果对于实体集 A 中的每一个实体,实体集 B 中有 n 个实体($n\geqslant 0$)与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中至多只有一个实体与之联系,则称实体集 A 与实体集 B 有一对多联系,记为1∶n。

例如，一个班级中有若干名学生，而每个学生只在一个班级中学习，则班级与学生之间具有一对多联系。

(3) 多对多联系($m:n$)

如果对于实体集 A 中的每一个实体，实体集 B 中有 n 个实体($n\geqslant0$)与之联系，反之，对于实体集 B 中的每一个实体，实体集 A 中也有 m 个实体($m\geqslant0$)与之联系，则称实体集 A 与实体集 B 具有多对多联系，记为 $m:n$。

例如，一名学生可以选修多门课程，一门课程可以被多名学生选修，学生集合与学校开设的课程集合之间存在 $m:n$ 联系。

2. 数据模型的分类

能表示实体类型及实体间联系的模型称为数据模型。数据模型的种类很多，目前被广泛使用的可分为两种类型。一种是独立于计算机系统的数据模型，它完全不涉及信息在计算机中的表示，这类模型称为"概念数据模型"。概念模型是按用户的观点对数据建模，易于用户理解，是对现实世界的第一层抽象，是用户和数据库人员进行交流的工具。这一类模型中最著名的是"实体联系模型"。另一种数据模型是直接面向数据库的逻辑结构，是对现实世界的第二层抽象，与数据库管理系统有关，称为"逻辑数据模型"，例如层次、网状、关系、面向对象等模型。这类模型有严格的形式化定义，以便于在计算机系统中实现。本章重点阐述关系模型。

6.3 实体联系模型

实体联系模型(Entity Relationship Model，ER 模型)是 P. P. Chen 于 1976 年提出的。这个模型直接从现实世界中抽象出实体类型及实体间联系，然后用实体联系图(ER 图)表示数据模型。ER 图是表示概念模型的有力工具。

ER 图有以下三个基本成分。

(1) 矩形框，用于表示实体类型(考虑问题的对象)。

(2) 菱形框，用于表示联系类型(实体间联系)。

(3) 椭圆形框，用于表示实体类型和联系类型的属性。

相应的命名均写在框中。实体与属性之间、联系与属性之间用直线连接；联系类型及其涉及的实体类型也以直线相连，用来表示它们之间的联系，并在直线端部标注联系的类型。例如，学生实体型及其属性 ER 图如图 6.2 所示。两个实体型 ER 图实例如图 6.3 所示。

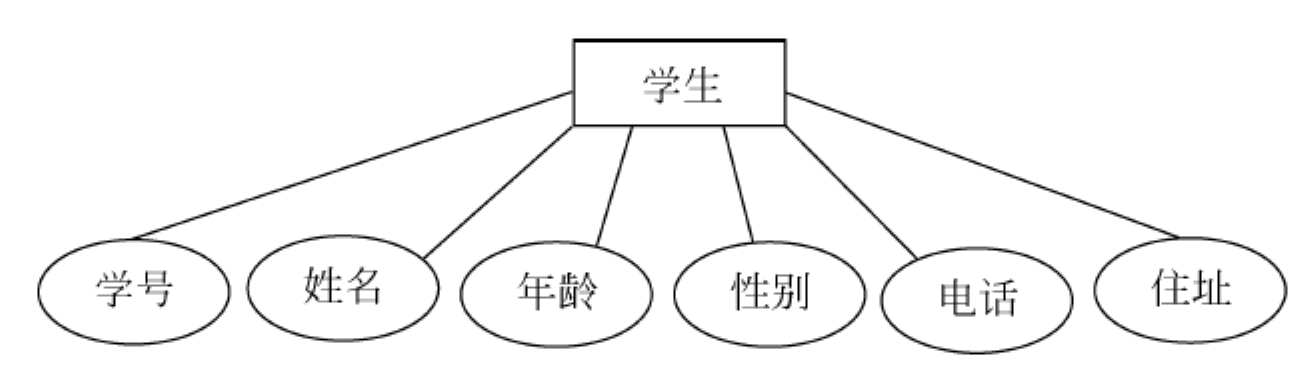


图 6.2　学生实体类型及属性 ER 图

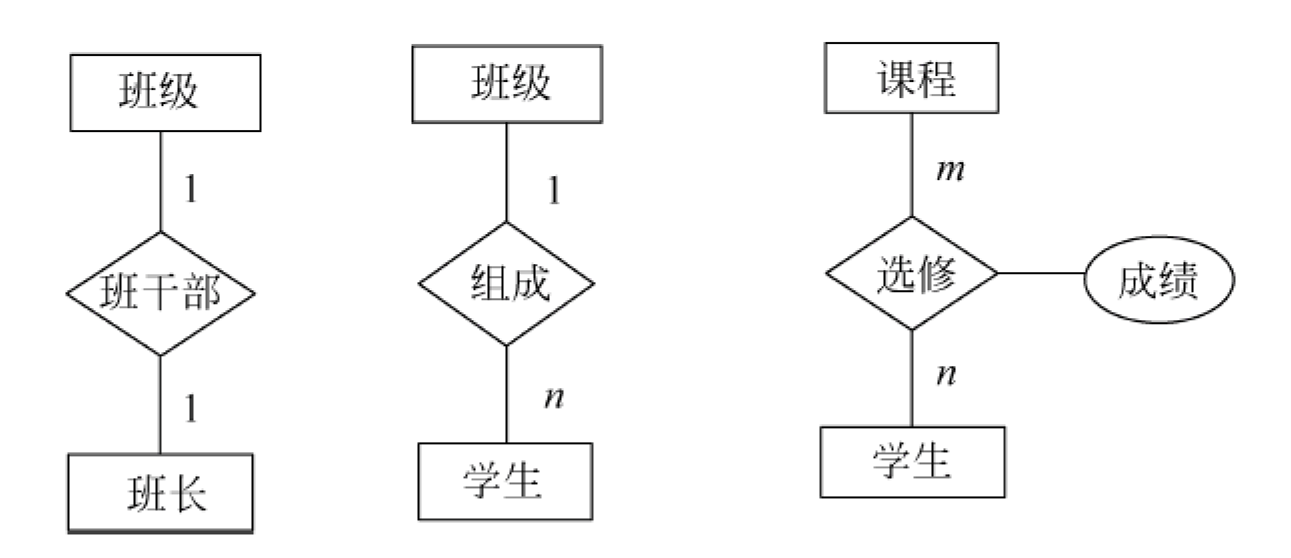


图 6.3 两个实体类型间的联系 ER 图

一般地，两个以上的实体类型之间也存在着一对一、一对多、多对多联系。例如，对于课程、教师与参考书三个实体类型，如果一门课程可以有若干个教师讲授，使用若干本参考书，每一个教师只讲授一门课程，每一本参考书只供一门课程使用，则课程、教师与参考书三者之间的联系是一对多的，如图 6.4 所示。

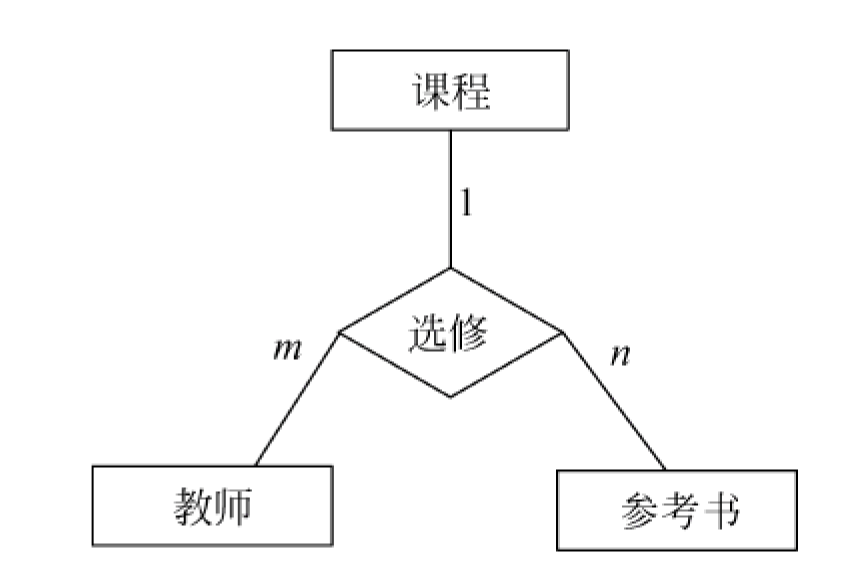


图 6.4 两个以上实体型间 1∶n 联系 ER 图

下面通过例子说明设计 ER 图的过程。

【例 6.1】 学生成绩管理子系统：一个系有若干名学生，一名学生只能在一个系学习；学校开设多门课程供学生选修，一名学生可选修多门课程，一门课程可供多名学生选修，为学生成绩管理子系统设计一个 ER 模型。ER 图建立过程如下。

(1) 首先确定实体类型。本题目有三个实体类型：学生 XS，系 DEPT，课程 KC。

(2) 确定联系类型。DEPT 和 XS 之间是 1∶n 联系，XS 和 KC 之间是 m∶n 联系。

(3) 把实体类型和联系类型组合成 ER 图。

(4) 确定实体类型和联系类型的属性。

学生(XS)属性：

```
学号(sno),姓名(sname),性别(sex),出生日期(bir),所在系编号(dno)
```

系(DEPT)属性：

```
系编号(dno),系名称(dname),系办公地点(dloc)
```

课程(KC)属性：

```
课程编号(cno),课程名称(cname),学期(term),学时(period),学分(crediit)
```

SC(XS 和 KC 之间 m∶n 联系)属性：

成绩(grade)

(5) 确定实体型的主键,在 ER 图中属于主键的属性名下画一条横线。

本例 ER 图如图 6.5 所示。

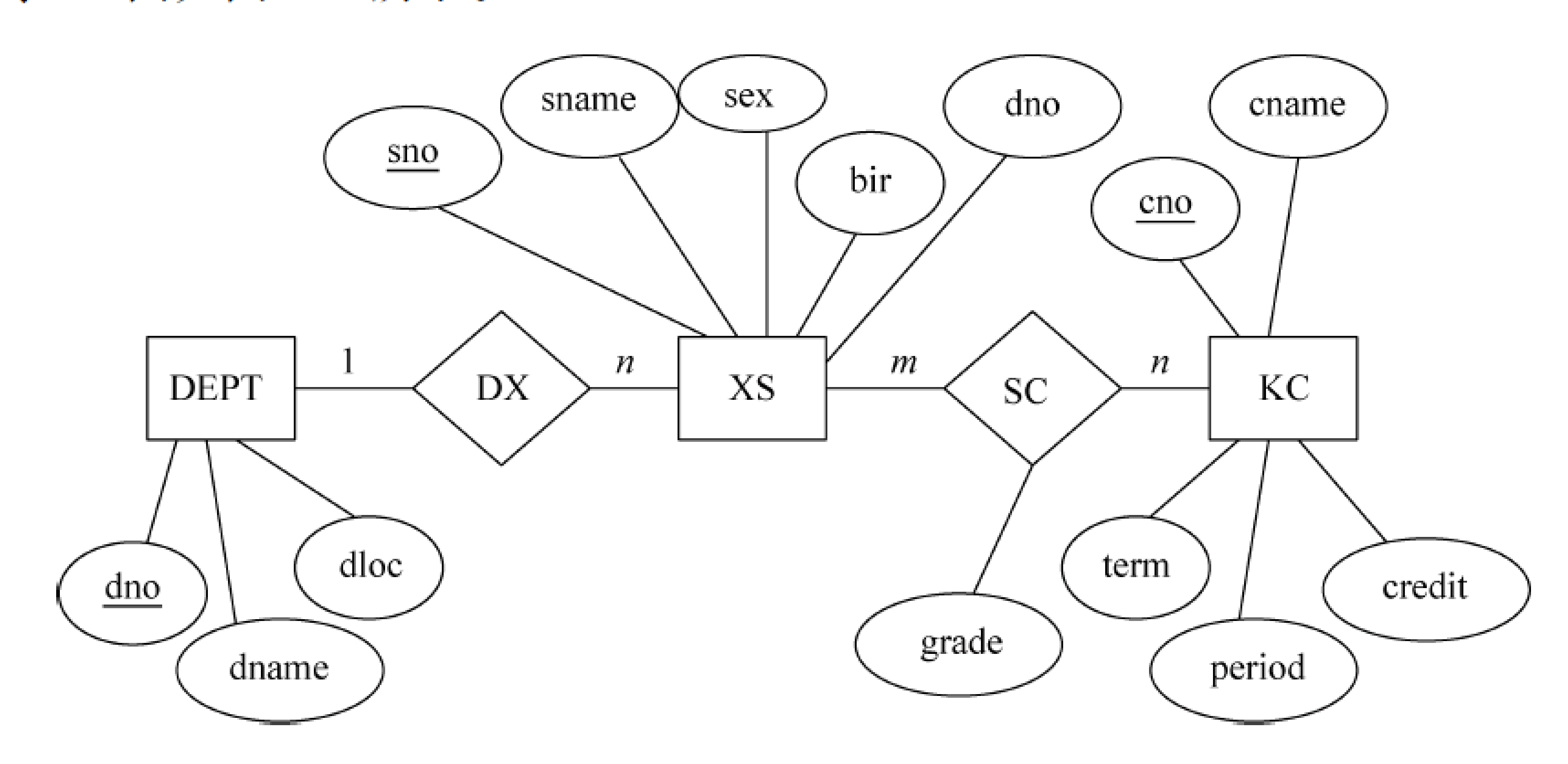


图 6.5　ER 图实例

6.4　关系模型

关系模型是以二维表格(关系表)的形式组织数据库中的数据,即用二维表格表示实体集。关系模型是由若干关系模式组成的集合。它的实例称为关系,每个关系实际上是一张二维表格。

如何将例 6.1 中的 ER 图转换成关系模型?转换的方法是将 ER 图中的实体类型和联系的类型分别转换成关系模式即可,在属性名下加一横线表示模式的键。1∶n 联系类型(例如 DEPT 和 XS)相应的关系模式是与 n 端对应的关系模式合并,即在 n 端的关系模式(即 XS 模式)的属性中加入另一个关系模式(DEPT 模式)的键和联系本身的属性;m∶n 联系类型相应的关系模式属性由联系类型属性和与之联系的实体类型的键一起组合而成。例 6.1 转换成的关系模式如下:

XS 模式　(sno,sname,sex,bir,dno)

DEPT 模式　(dno,dname,dloc)

KC 模式　(cno,cname,term,period,credit)

SC 模式　(sno ,cno,grade)

如图 6.6 所示为具体实例,即具体的三个关系。

1. 关系模型的基本概念

1) 关系

关系可以看成是由行和列交叉组成的二维表格,它表示的是一个实体集合。

(1) 表中一行称为一个元组,可用来表示实体集中的一个实体。

(2) 表中的列称为属性,给每一列起一个名称即属性名,表中的属性名不能相同。

XS 关系：

sno	sname	sex	bir	dno
130601	赵林林	女	1995-2-10	070001
130602	李芳	女	1994-10-3	070002
130603	张强	男	1995-8-8	070001
130604	王彦军	男	1994-6-10	070001
130605	罗琳	女	1994-11-1	070001

DEPT 关系：

dno	dname	dloc
070001	计算机科学与技术系	11 号楼 12 层 1201
070002	网络工程系	11 号楼 12 层 1202
070003	信息技术系	11 号楼 12 层 1203

KC 关系：

cno	cname	term	period	credit
101	计算机文化基础	1	64	4
102	面向对象程序设计	2	64	4
103	数据结构	3	64	4
104	操作系统	3	48	3
105	数据库原理	4	64	4
106	离散数学	3	72	6

SC 关系：

sno	cno	grade
070001	101	85
070002	101	90
070003	101	87
070001	102	90
070002	102	70

图 6.6 关系模型实例

(3) 列的取值范围称为域，同列具有相同的域。例如，年龄为整数域。

(4) 表中任意两行(元组)不能相同。能唯一标识表中不同行的属性或属性组称为主键。

尽管关系与二维表格传统的数据文件有类似之处，但它们又有区别，严格地说，关系是一种规范化了的二维表格，具有如下性质。

(1) 列是同质的：每一列中的分量是同一类型的数据，来自同一域。

(2) 不同列可来自同一个域：不同列(属性)要给予不同的属性名。

(3) 列的顺序无所谓：列的次序可以任意交换。

(4) 任意两个元组不能完全相同。

(5) 行的顺序无所谓：行的次序可以任意交换。

(6) 分量必须取原子值：每一个分量都必须是不可分的数据项。

2）关系模式

关系模式是对关系的描述，它包括模式名，组成该关系的诸属性名、值域名和属性间数据的依赖。关系模式通常可以简记为：关系名(属性名 1，属性名 2，…)。

例如，ER 图表示的学生情况的部分转换成相应的关系模式为：

XS(sno,sname,sex,bir,dno)

关系模式 XS 描述了学生实体集的数据结构。其中，XS 为关系名，组成该关系的属性名包括：sno(学号)，sname(姓名)，sex(性别)，bir(出生日期)，dno(所在系)。

3）键

在关系数据库中，键也称码，是关系模型的一个重要概念。通常键由一个或几个属性组成，键有如下几种类型。

(1) 候选键：如果一个属性集能唯一标识元组，且又不含有多余的属性，那么这个属性集称为关系的候选键。候选键中的诸属性称为关系的主属性。

(2) 主键：若一个关系中有多个候选键，则选其中的一个为关系的主键。

(3) 外键：关系模式 R1 中的某一属性(或属性组)F 与关系模式 R2 的主键相对应，但不是 R1 的码，则称 F 是关系模式 R1 的外键。需要说明的是，R1、R2 不一定是不同的关系模式，也可以是同一关系模式；外键不一定要与相对应的主键同名，只需定义在相同的值域上即可。在实际应用中，为了便于辨识，当外键与相对应的主键位于不同关系模式时，通常会给它们取相同的名字。其中，基本关系 R1 称为参照关系，基本关系 R2 称为被参照关系或目标关系。

例如，学生关系模式 XS 和系关系模式 DEPT 分别如下。

XS(sno，sname，sex，bir，dno)：学生关系模式的属性分别为学号、姓名、性别、年龄和学生所在系编号。

DEPT(dno，dname，dloc)：系关系模式的属性分别为系编号、系名称、系办公地点。

学生关系模式的主键是 sno，系关系模式的主键为 dno，在学生关系模式中，dno 是它的外键。学生关系为参照关系，系关系为被参照关系。在关系数据库中，表与表之间的联系是通过外键(即公共属性)实现的。

2. 关系模型的完整性规则

关系模型中有三类完整性规则，包括实体完整性、参照完整性和用户自定义完整性。

1）实体完整性规则

若属性(指一个或一组属性)A 是基本关系 R 的主属性，则属性 A 不能取空值。

【例 6.2】 在学生成绩管理数据库中，关系模式分别为：

XS (学号，姓名，性别，出生日期，所在系)

KC(课程号，课程名，学期，学时，学分)

SC (学号，课程号，成绩)

在学生关系 XS 中，"学号"为主键，按照实体完整性规则，则它不能取空值。选课关系中，"学号"和"课程号"为主键，则"学号"和"课程号"两个属性都不能取空值。

2）参照完整性规则

若属性(或属性组)F 是基本关系 R 的外键，它与基本关系 S 的主键 Ks 相对应(基本关

系 R 和 S 不一定是不同的关系),则对于 R 中每个元组在 F 上的值必须为:或者取空值(F 的每个属性值均为空值);或者等于 S 中某个元组的主键值。

【例 6.3】 学生关系模式和系关系模式如下,其中主键用下划线标识:

XS (学号,姓名,性别,出生日期,所在系编号)

DEPT(系编号,系名称,系办公地点)

这两个关系之间存在着属性的引用,即学生关系引用了专业关系的主键“系编号”。显然,学生关系中的“所在系编号”值必须是确实存在的系编号,即系关系中应该有该系的记录。这也就是说,学生关系中的某个属性的取值需要参照系关系中某个属性的取值。学生关系的“所在系编号”与系关系的“系编号”相对应,因此,“所在系编号”属性是学生关系的外键,对应的“系编号”是系关系的主键。这里系关系是被参照关系,学生关系为参照关系。

对于例子中学生关系中的每个元组的“所在系编号”属性只能取下面两类值:空值,表示尚未给该学生分配系;非空值,该值必须是系关系中某个元组的“系编号”的值,表示该学生不可能分配到一个不存在的系中。即被参照关系“系”中一定存在一个元组,它的主键值等于该参照关系“学生”中的外键值。

3) 用户定义的完整性规则

在建立关系模式时,对属性定义了数据类型,即使这样可能还满足不了用户的需求,此时,用户可以针对具体应用设置完整性规则。即用户自定义的完整性规则,它反映某一具体应用所涉及的数据值必须满足的语义要求。例如,某个属性必须取唯一值,某个属性的取值范围在 0~100(例如学生的成绩)之间。

使用关系模型的数据库就是关系型数据库。小型的关系型数据库管理系统包括 MySQL、Access 等。

小结

本章概述了数据库的基本概念,数据模型是数据库系统的核心和基础。简要介绍了概念模型及关系模型,概念模型用于信息世界的建模,ER 模型是这类模型的典型代表。ER 模型方法简单、易于理解,应用十分广泛。关系模型是目前最重要的一种数据模型,关系数据库系统采用关系模型作为数据的组织方式。本书所介绍的 MySQL 数据库管理系统是一个小型关系型数据库管理系统。本章概念较多,如果是刚开始学习数据库,可在学习后面章节后再回来理解和掌握这些概念。

习题

1. 什么是数据、数据库、数据管理系统、数据库系统?
2. 简述概念模型、关系模型。
3. 解释概念模型中以下术语:实体、实体型、实体集、属性、码、实体-联系图。
4. 解释关系模型中以下术语:关系、属性、域、元组、码、分量、关系模式。
5. 简述关系模型的完整性规则。

第7章 MySQL数据库基础

本章要点：

- MySQL 下载；
- MySQL 服务器的安装与配置；
- MySQL 的卸载；
- MySQL 软件的常用操作；
- MySQL 官方客户端软件；
- MySQL 常用图形化管理软件 SQLyog 软件。

随着数据库技术的发展，一些具有免费使用、配置简单、稳定性好、性能优良等特点的开源数据库管理系统逐渐流行起来，且在中低端应用中占据很大的市场份额。本书所介绍的 MySQL 数据库管理系统是一个小型关系型数据库管理系统，且是开源数据库的杰出代表。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。且由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

7.1 MySQL 下载

对于不同的操作系统，MySQL 提供了相应的版本。本节将以 Windows 平台下的图形化安装包为例，详细介绍 MySQL 的下载、安装和配置等过程。

1. MySQL 的各种版本

目前 MySQL 数据库按照用户群分为社区版和企业版，这两个版本的重要区别为：社区版可以自由下载且完全免费，但是官方不提供任何技术支持，适用于大多数普通用户；企业版收费，但是版本提供了更多的功能，可以享受完备的技术支持，适用于对数据库的功能和可靠性要求比较高的企业客户。

MySQL 版本更新非常快，常见的软件版本有 GA（官方推崇广泛使用的版本）、RC（候选版本，该版本最接近正式版的版本）、Alpha 和 Bean（这两种版本属于测试中版本，其中 Alpha 是指内测版本，指开发团队内部测试的版本或者有限用户体验测试版本；Bean 是指公测版本，针对所有用户公开的测试版本）。从 MySQL 版本 5 开始支持触发器、视图、存储过程等高级特性。

2. 下载 MySQL 软件

MySQL 软件是完全网络化的跨平台关系型数据库,本教材下载的版本为 mysql-5.6.19,可以通过下面的步骤来实现该平台的下载,具体如下。

(1) 首先访问 MySQL 的官方网站(http://www.mysql.com/),如图 7.1 所示。

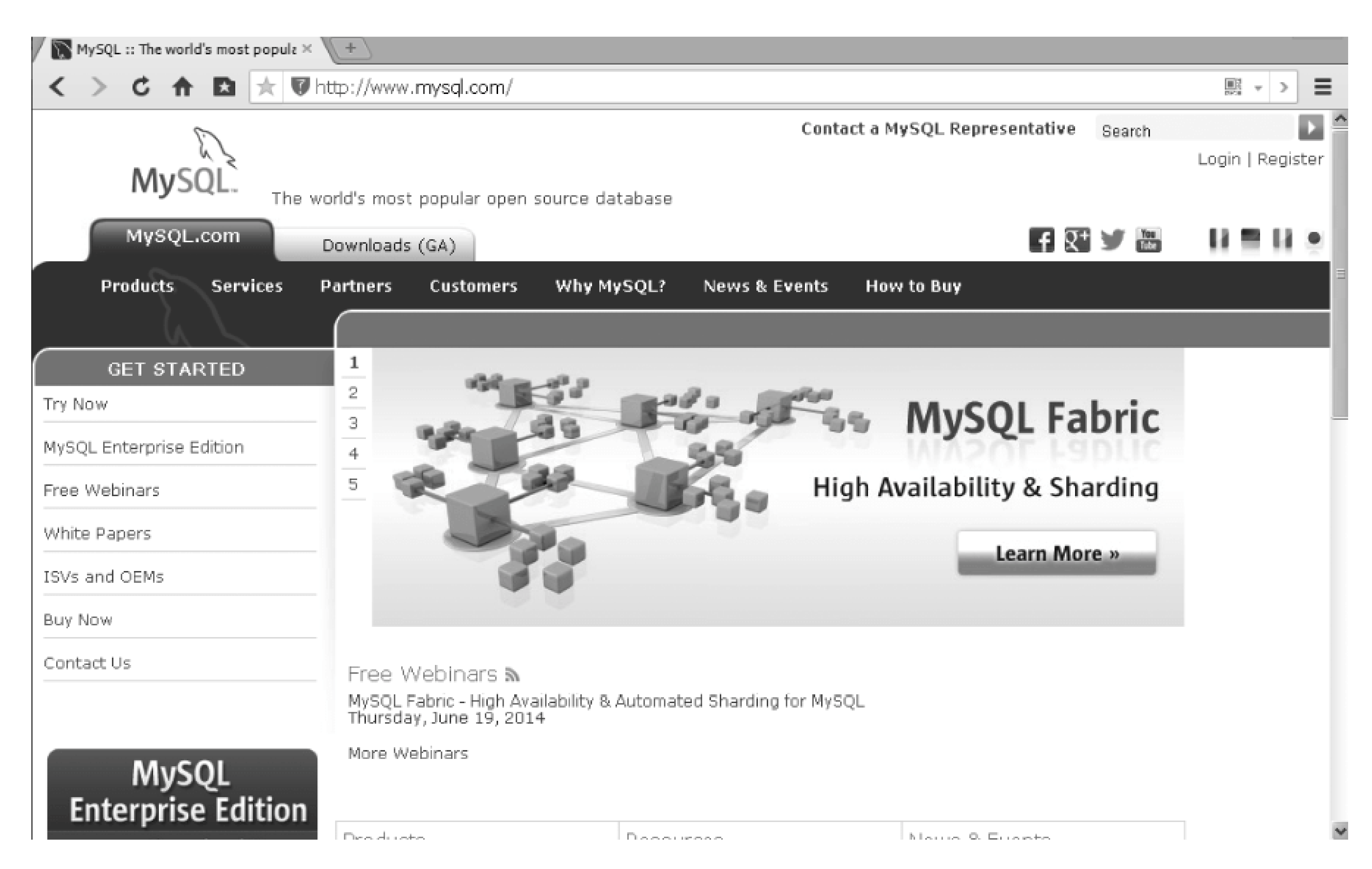


图 7.1 MySQL 官方网站首页

(2) 打开 MySQL 软件首页后,单击页面 Downloads(GA)导航栏,就会进入 MySQL 产品页面,如图 7.2 所示。在该页面单击社区版(MySQL Community Server)中的 Download 超链接,就会进入下载页面。

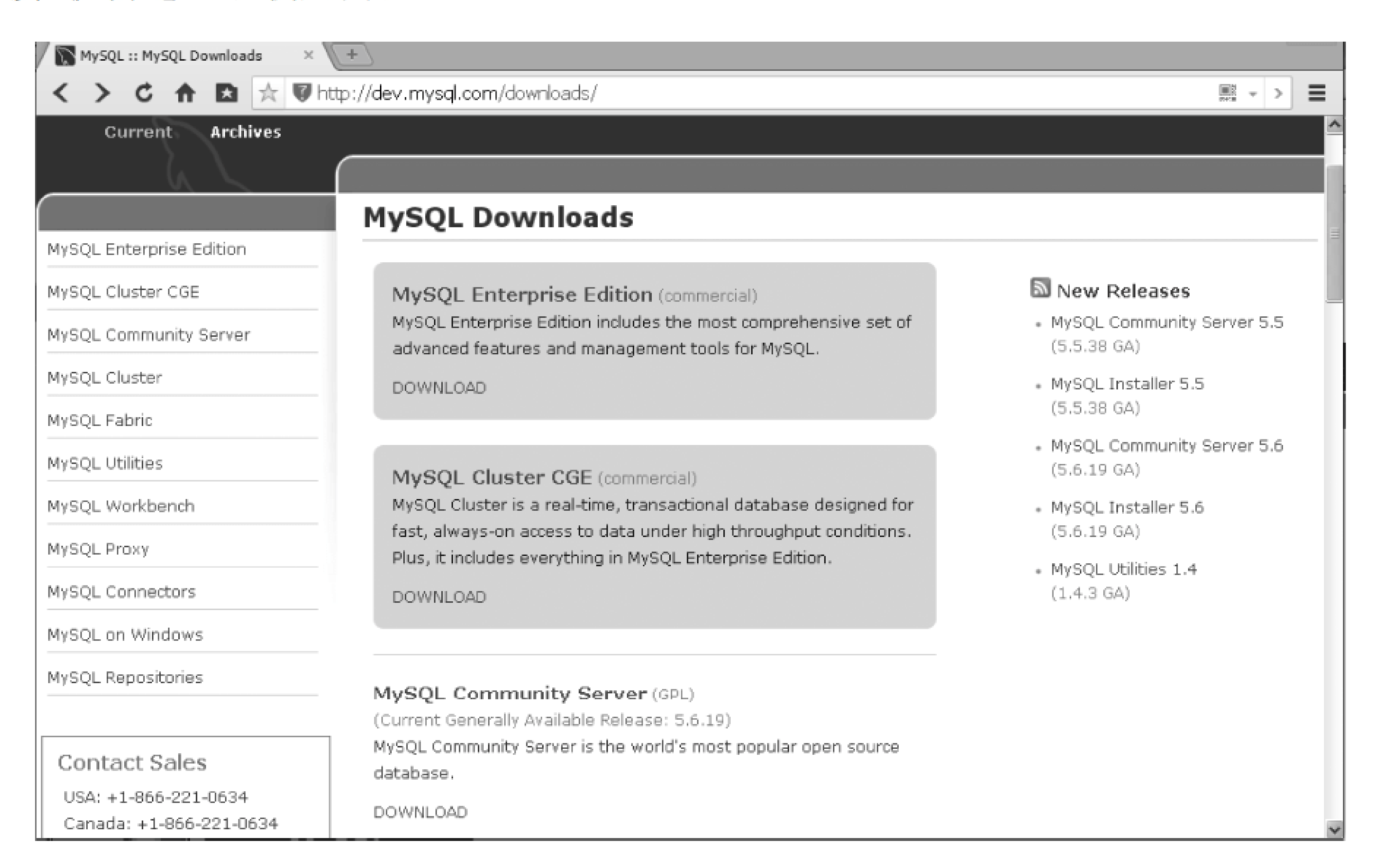


图 7.2 下载社区版

（3）在下载页面中首先选择 Select Platform 下拉菜单，在该下拉菜单中选择 Microsoft Windows 平台，然后单击 Windows（x86，32-bit），MSI Installer 选项右边的 Download 按钮来下载 32 位的 MySQL 安装软件，如图 7.3 所示。

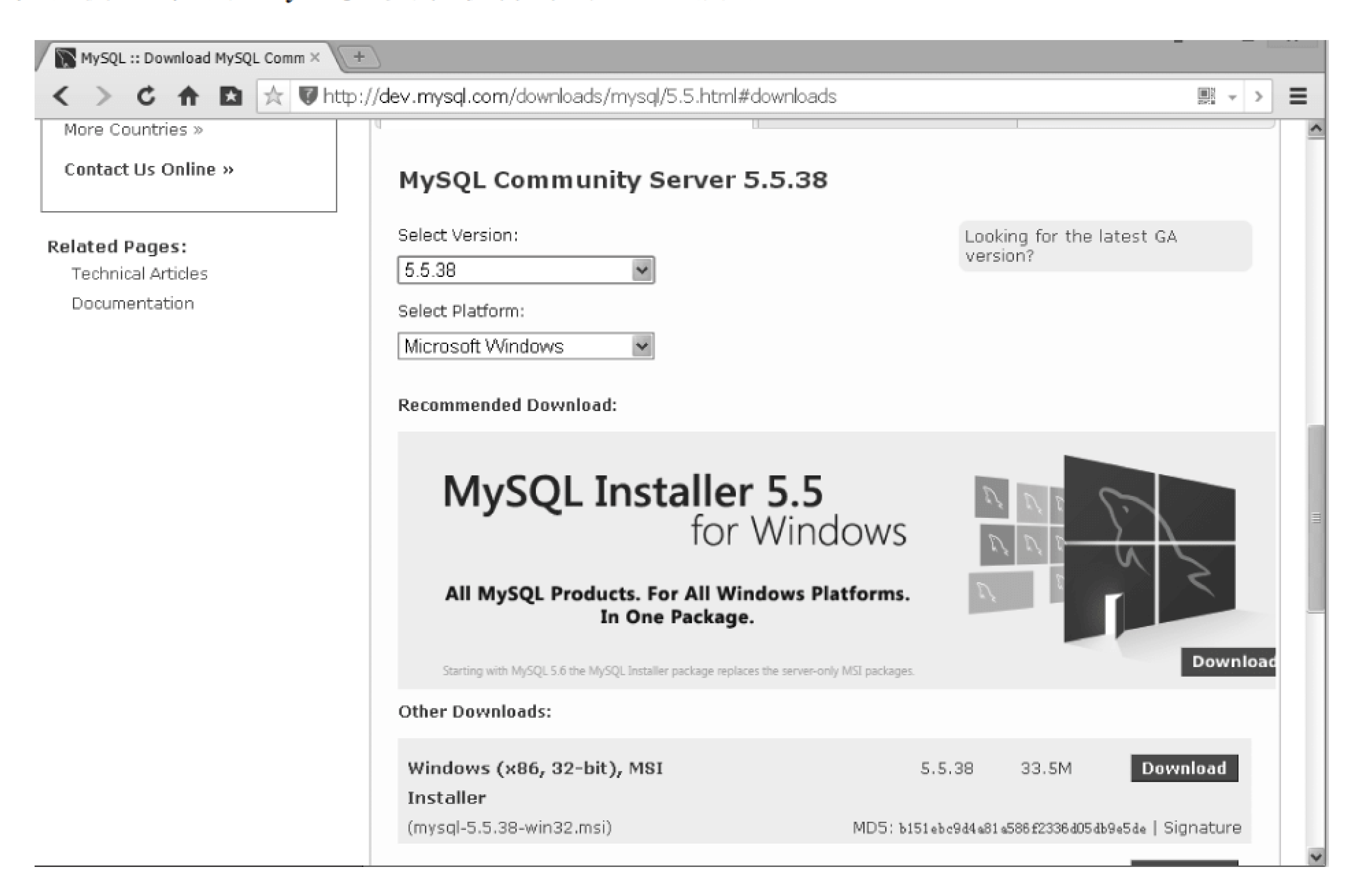


图 7.3　选择 MySQL 5.5.38

（4）选择相应平台和版本的 MySQL 软件后，当单击 Download 按钮后就会进入下载的页面。下载完 mysql-5.5.38 后，就可以安装该数据库软件了。

7.2　MySQL 服务器的安装与配置

1. MySQL 服务器的安装

下载完数据库 MySQL 软件的安装程序后，就可以开始安装该数据库了，具体的安装步骤如下。

（1）双击 MySQL 安装程序（mysql-5.5.38-win32.msi），接着会使用 Windows Intaller 开始安装过程，如图 7.4 所示。

（2）单击 Next 按钮后，在出现的对话框中选择 I accept the terms in the License Agreement 选项接受协议（如图 7.5 所示），然后单击 Next 按钮进入 Choose Setup Type 对话框。

（3）在 Choose Setup Type 对话框中，选择 Typical 单选按钮（如图 7.6 所示），就会进入 Ready to install MySQL Server 5.5 对话框。

（4）在 Ready to install MySQL Server 5.5 对话框中确认一下安装的信息，单击 Install 按钮开始对 MySQL 软件的安装，如图 7.7 所示。

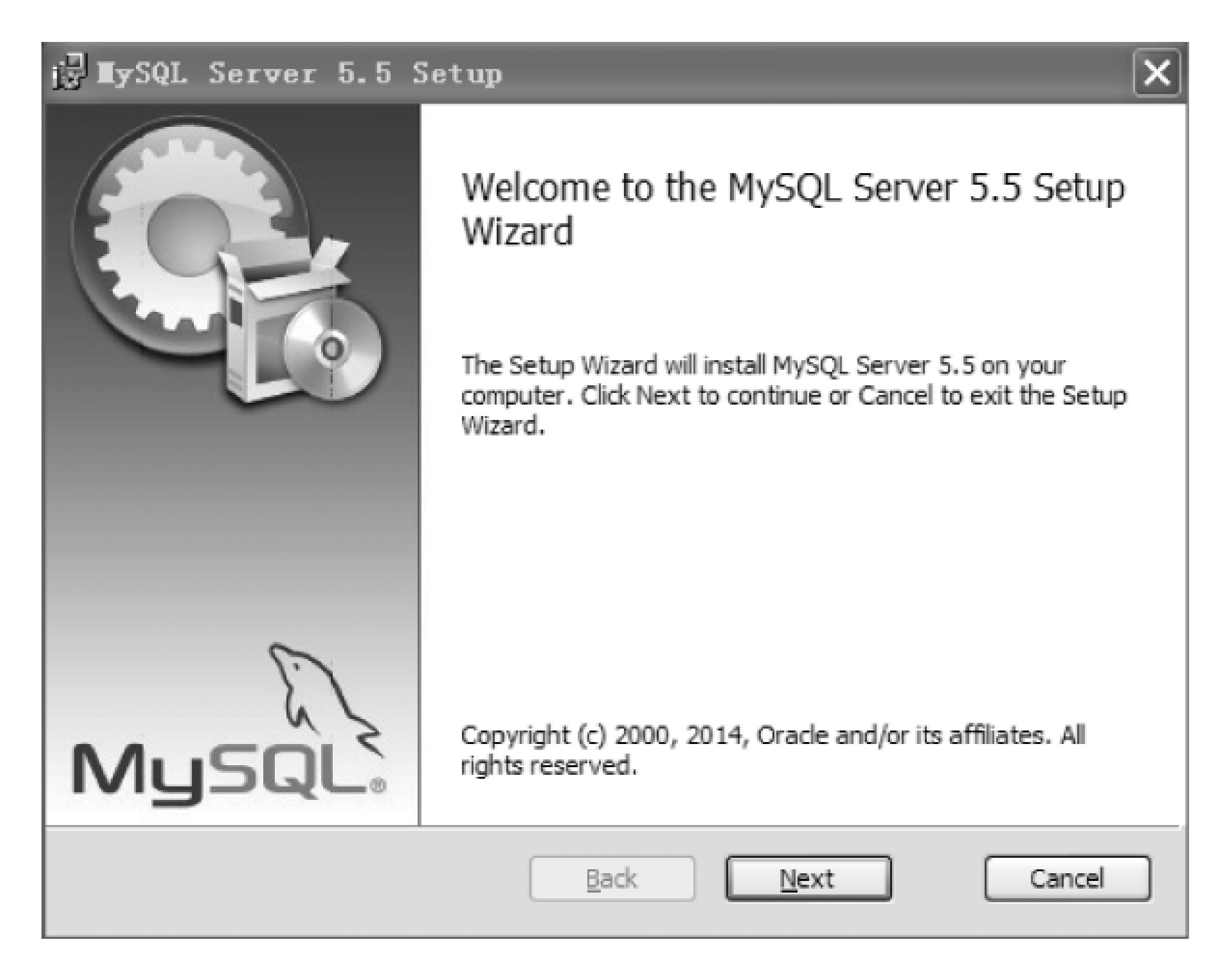


图 7.4　MySQL 安装欢迎界面

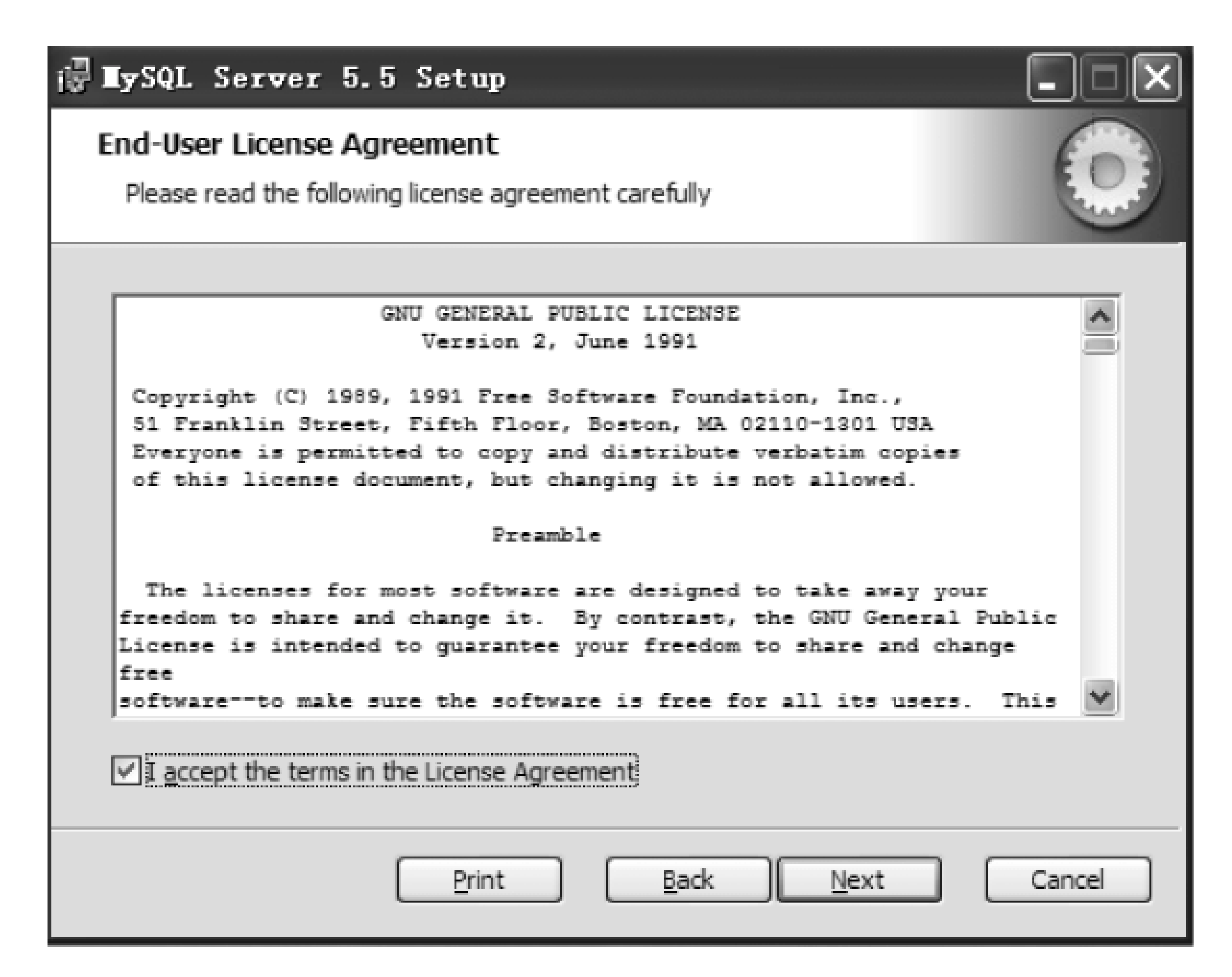


图 7.5　接受许可协议

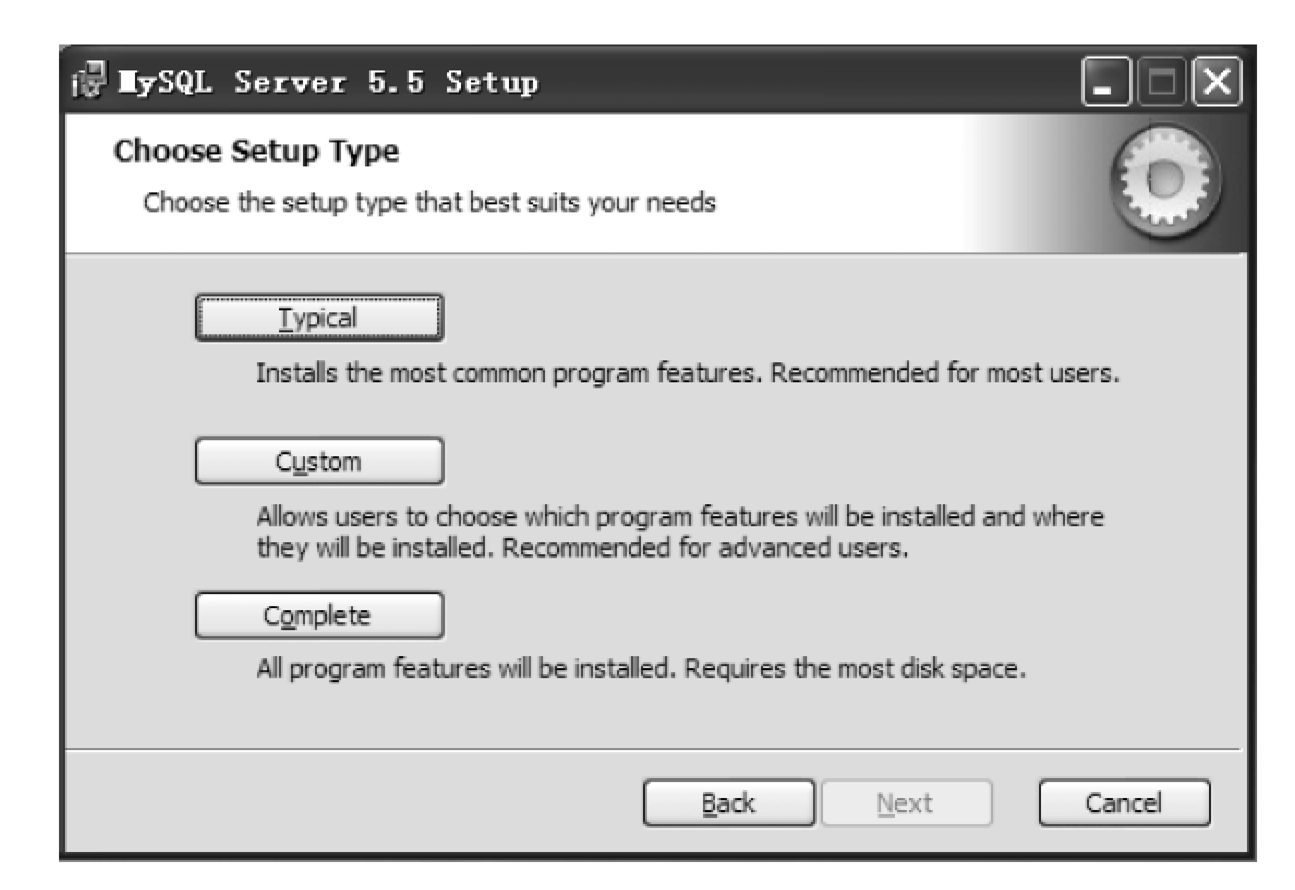


图 7.6　MySQL 安装类型

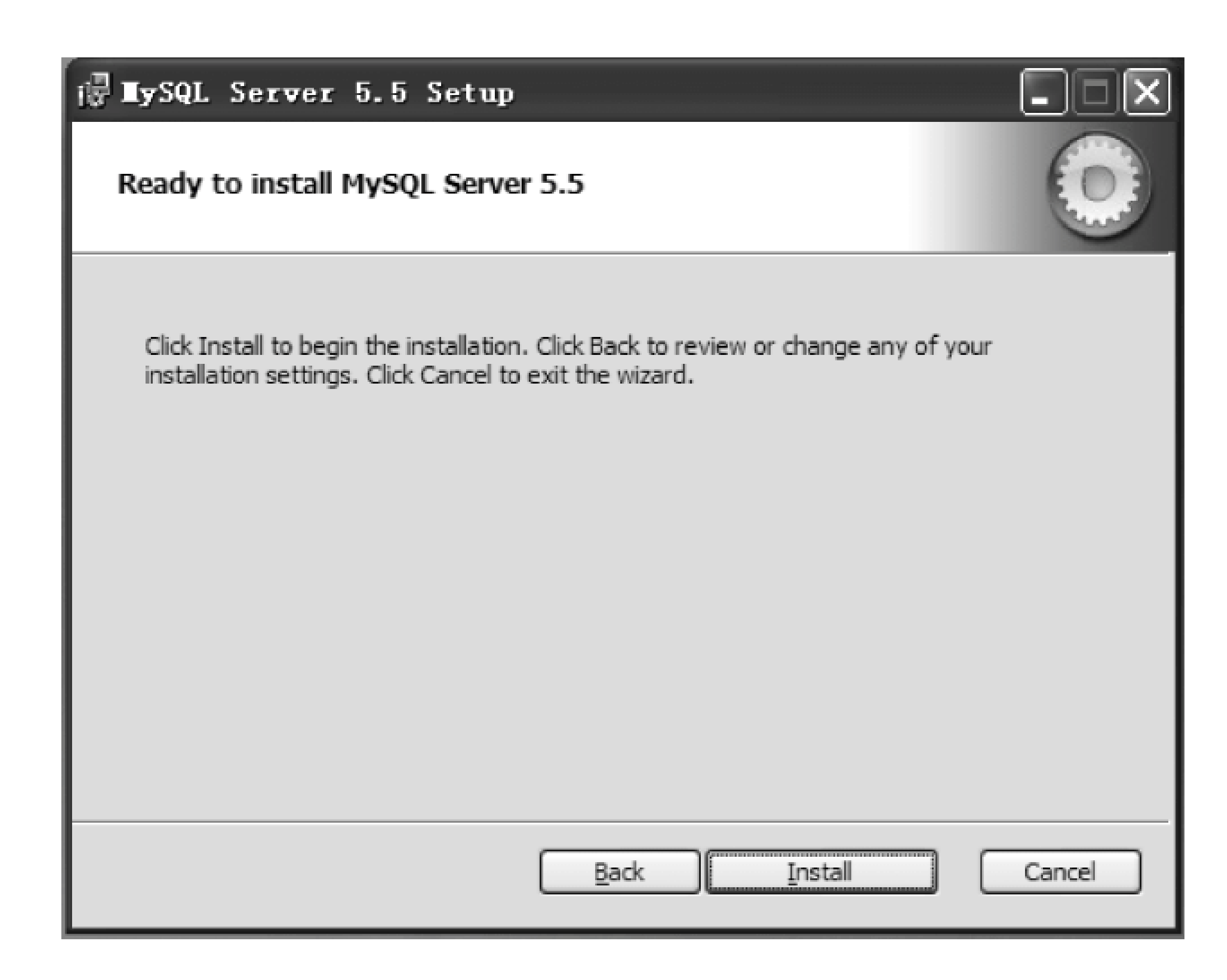


图 7.7　安装信息的确认

(5) 安装完成后，单击 Next 按钮会进入一系列说明界面，如图 7.8 所示。

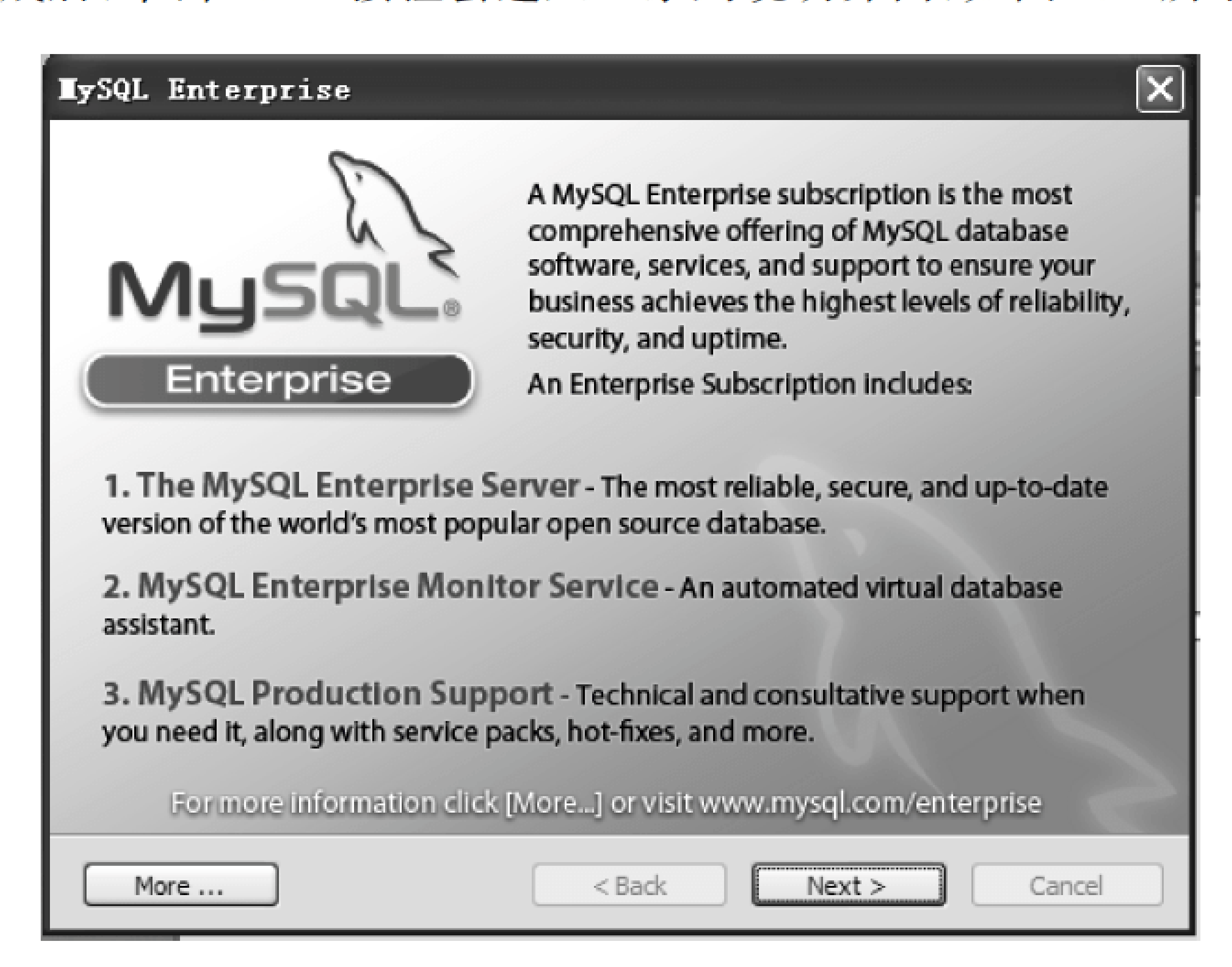


图 7.8　安装说明界面

(6) 最后进入安装完成的界面，在该对话框中会询问是否进行配置，如图 7.9 所示。如果不想现在进行配置，可以取消选中的 Launch the MySQL Instance Configuration Wizard 复选框，然后单击 Finish 按钮就会完成对 MySQL 软件的安装。

2. MySQL 服务器的配置

(1) 安装完毕后选择 Launch the MySQL Instance Configuration Wizard 复选框，单击 Finish 按钮进入配置向导的欢迎界面，单击界面中的 Next 按钮进入选择配置类型界面，如图 7.10 所示，配置类型有两种：Detailed Configuration(详细配置)和 Standard Configuration(标

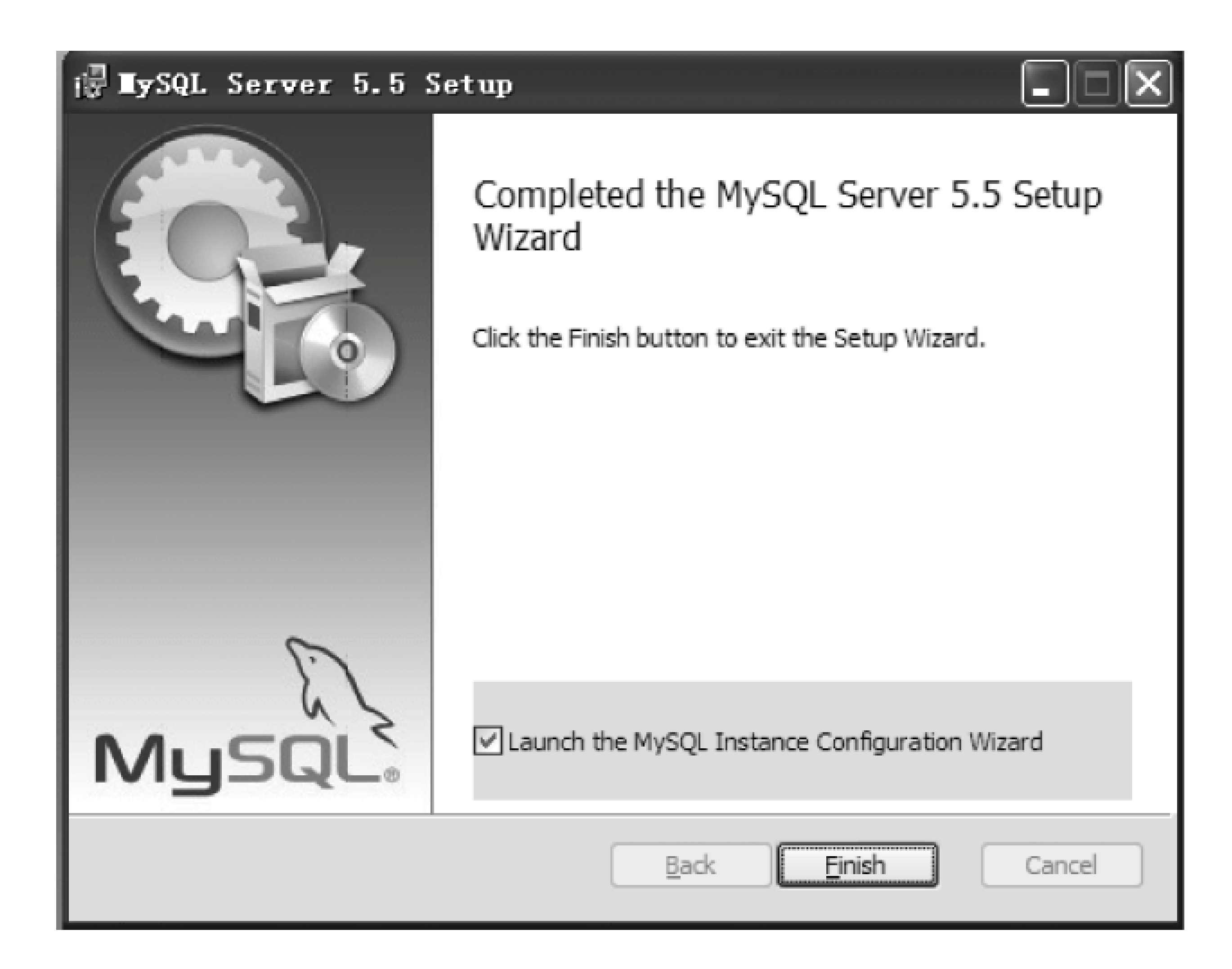


图 7.9 安装成功界面

准配置)。详细配置选项适合想要更加细粒度控制服务器配置的高级用户,标准配置选项适合想要快速启动 MySQL 而不必考虑服务器配置的新用户。

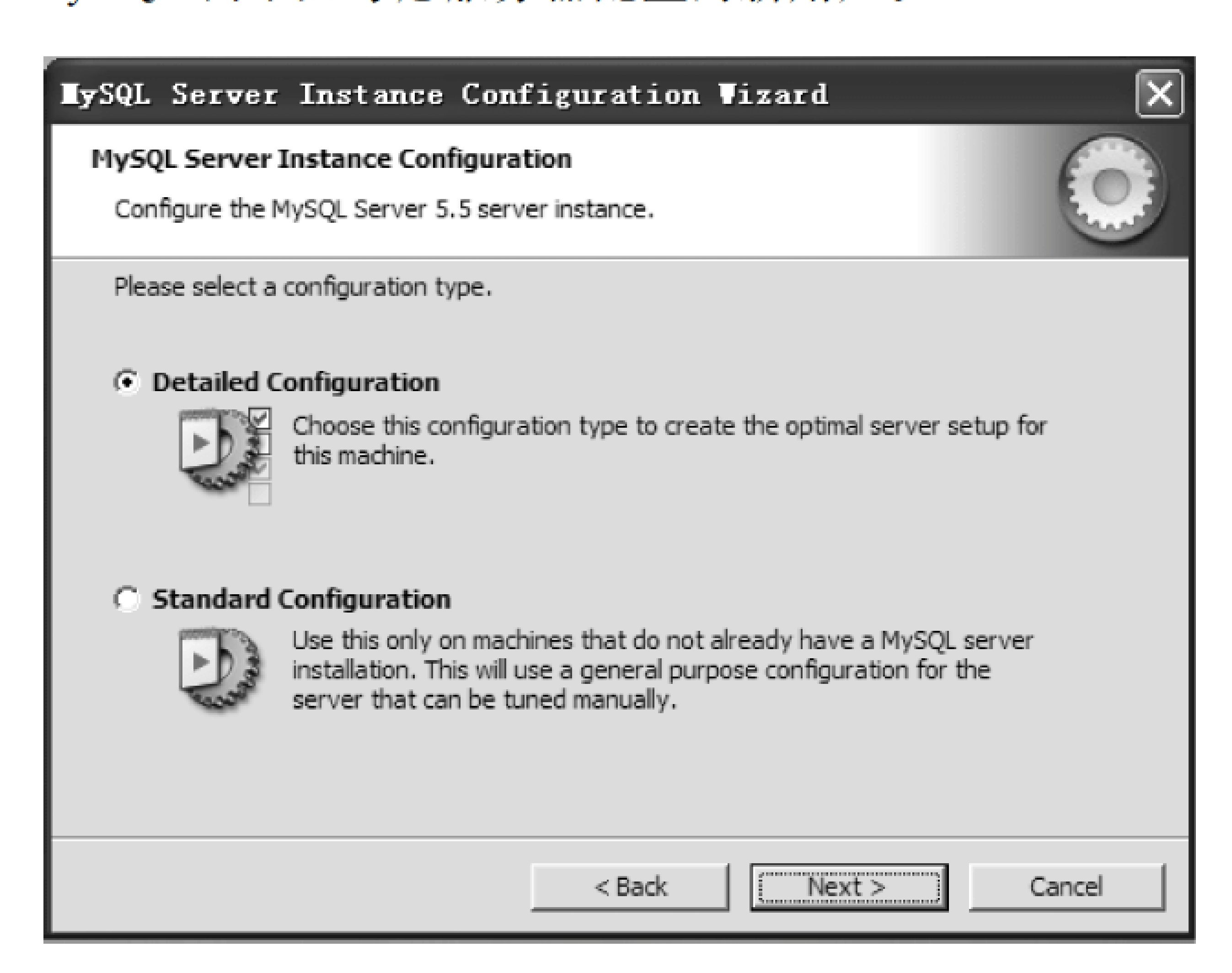


图 7.10 选择配置类型界面

(2) 在 MySQL 选择配置类型界面中,如图 7.11 所示,选择 Detailed Configuration 选项,单击 Next 按钮进行服务器类型选择,分为三种:Developer Machine(开发机器)、Server Machine(服务器)和 Dedicated MySQL Server Machine(专用 MySQL 服务器)。

(3) 在 MySQL 选择应用类型界面中选择 Developer Machine,然后单击 Next 按钮进入选择用途类型界面,如图 7.12 所示,有三个选项:Multifunctional Database(多功能数据库)、Transactional Database Only(事务性数据库)和 Non-Transactional Database Only(非事务性数据库)。

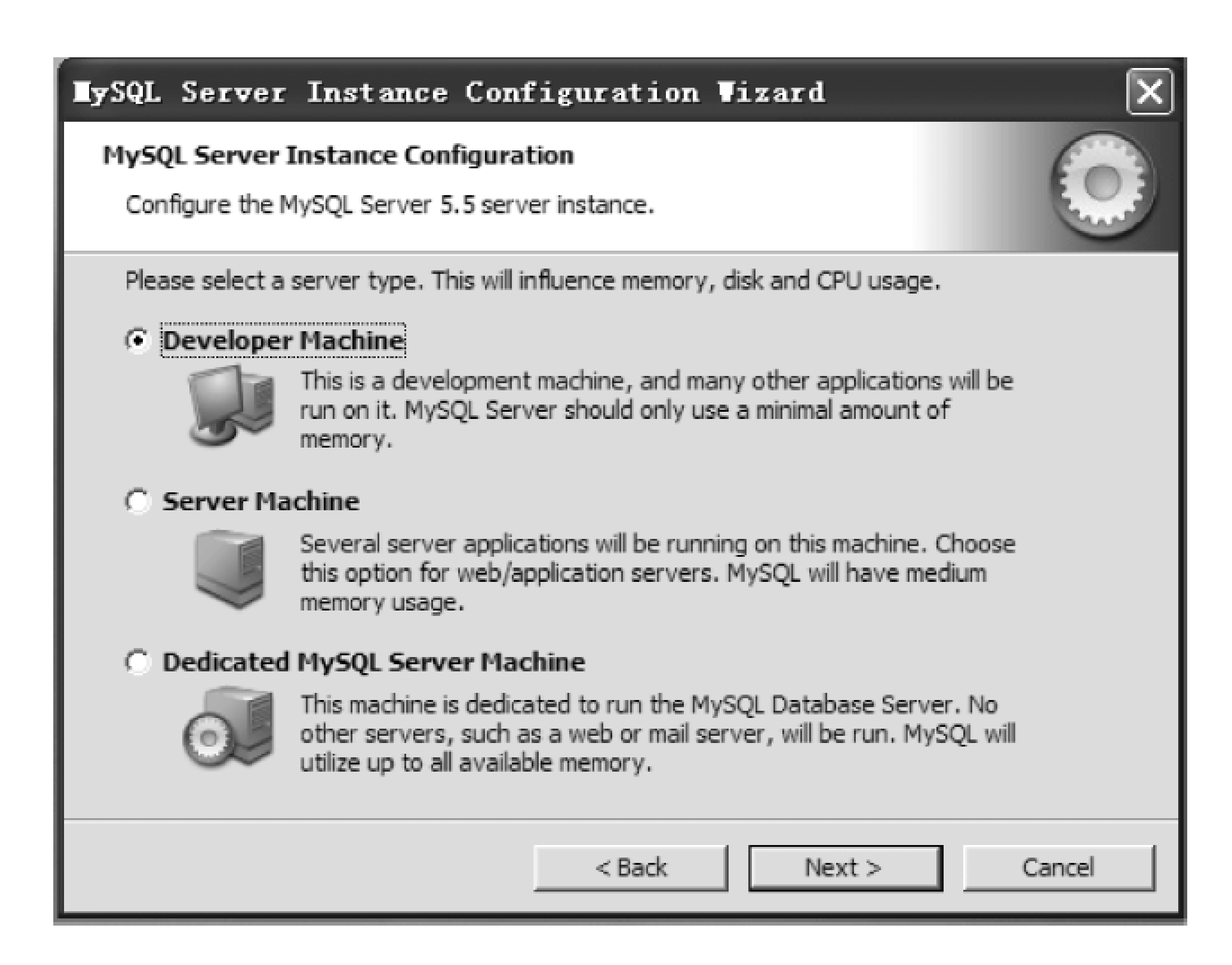


图 7.11 选择应用类型界面

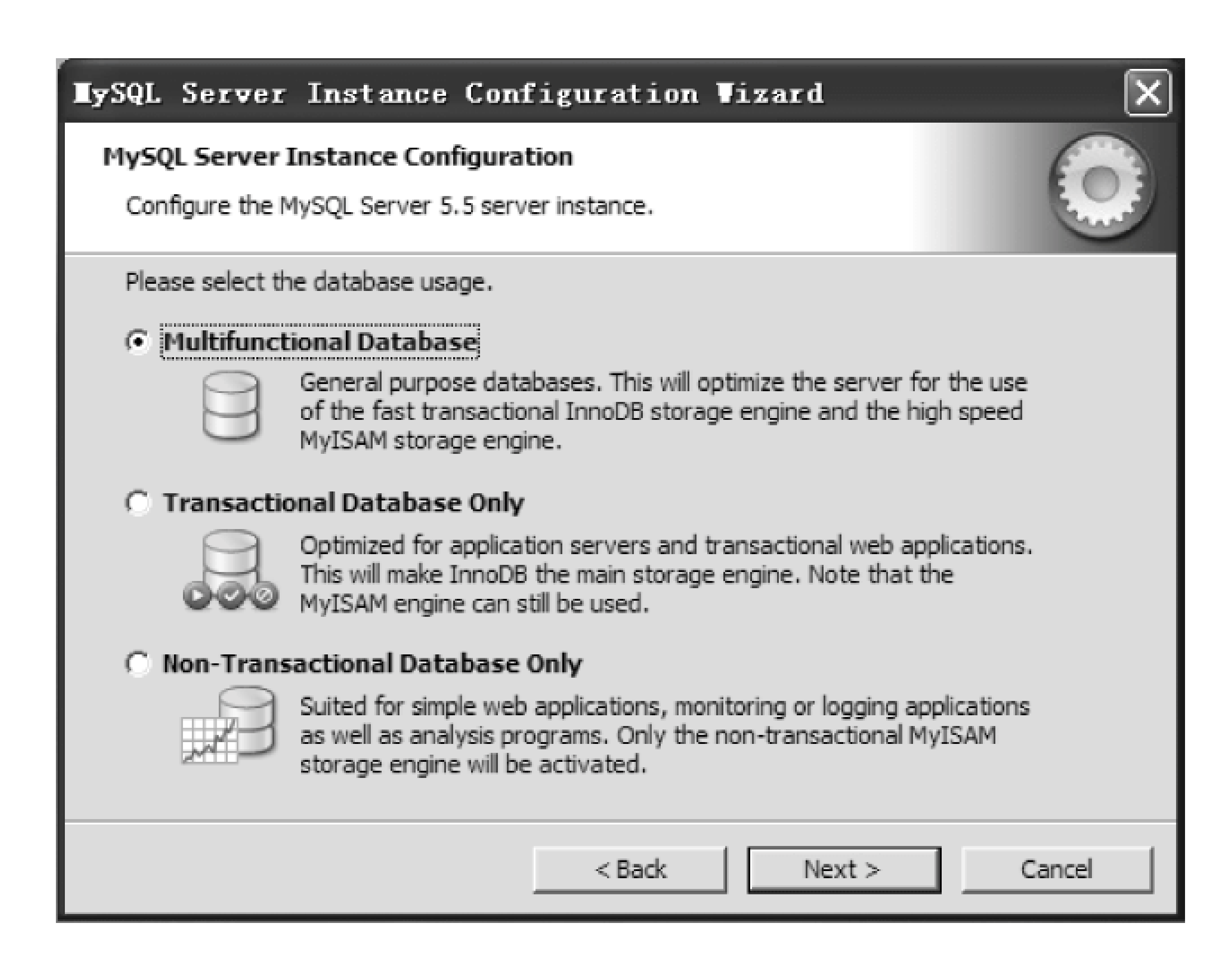


图 7.12 选择用途类型界面

(4) 在 MySQL 选择用途类型界面中选择 Multifunctional Database，然后单击 Next 按钮进入 InnoDB 数据文件目录配置界面，如图 7.13 所示。InnoDB 数据文件会在数据库第一次启动的时候创建，默认位置是在 MySQL 的安装目录下。用户可以根据实际的硬盘空间状况进行路径的选择。

(5) 在 InnoDB 数据文件目录里配置界面中保留默认值，然后单击 Next 按钮进入并发连接设置界面，如图 7.14 所示。MySQL 提供了三种类型并发处理供选择，Decision Support (DSS)/OLAP(决策支持)，如果服务器不需要大量的并行连接可以选择该选项；Online Transaction Processing(OLTP，联机事务处理)，如果服务器需要大量的并行连接则选择该选项；Manual Setting(手工设置)，选择该选项可以手动设置服务器并行连接的最大数目。

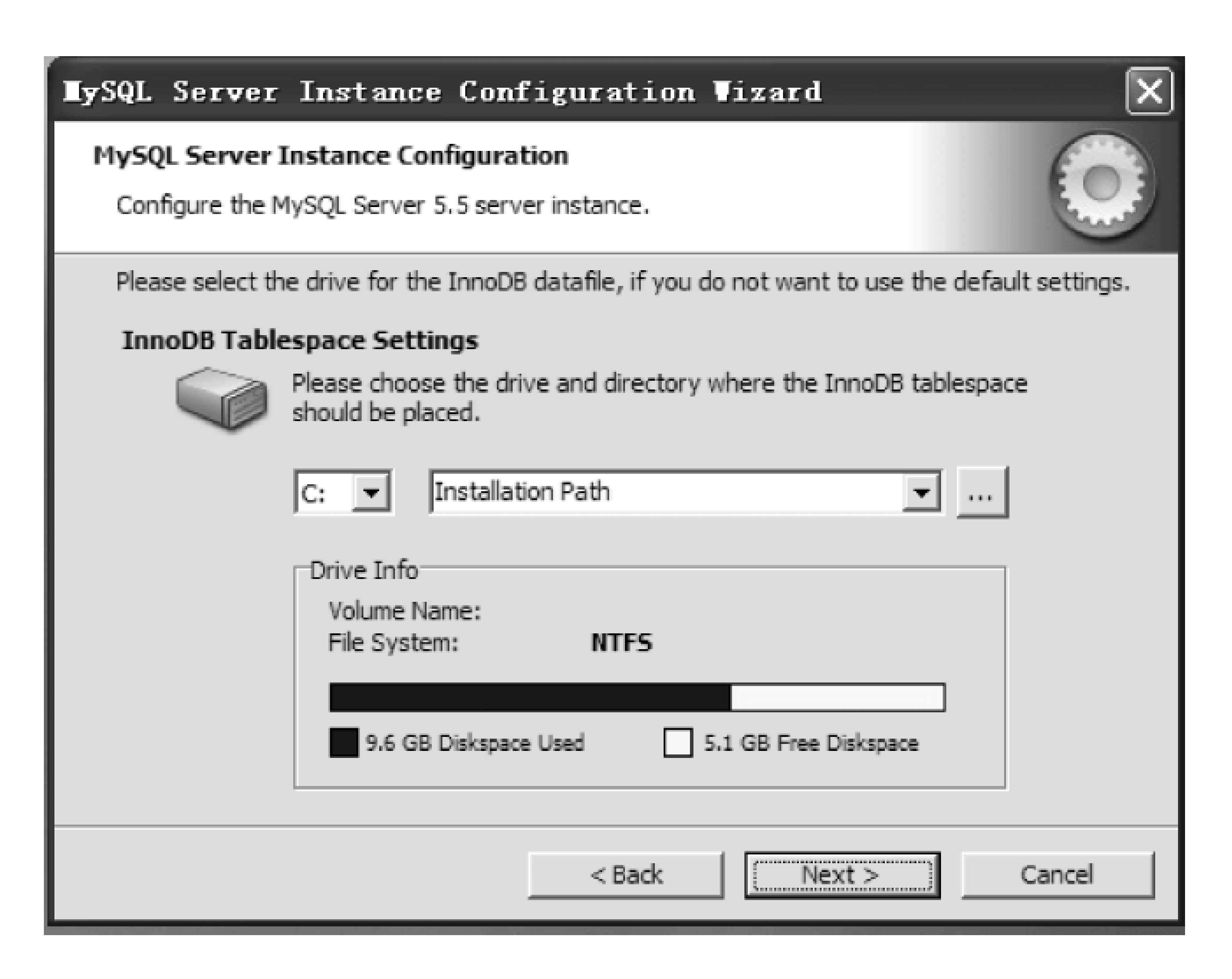


图 7.13 数据文件目录配置界面

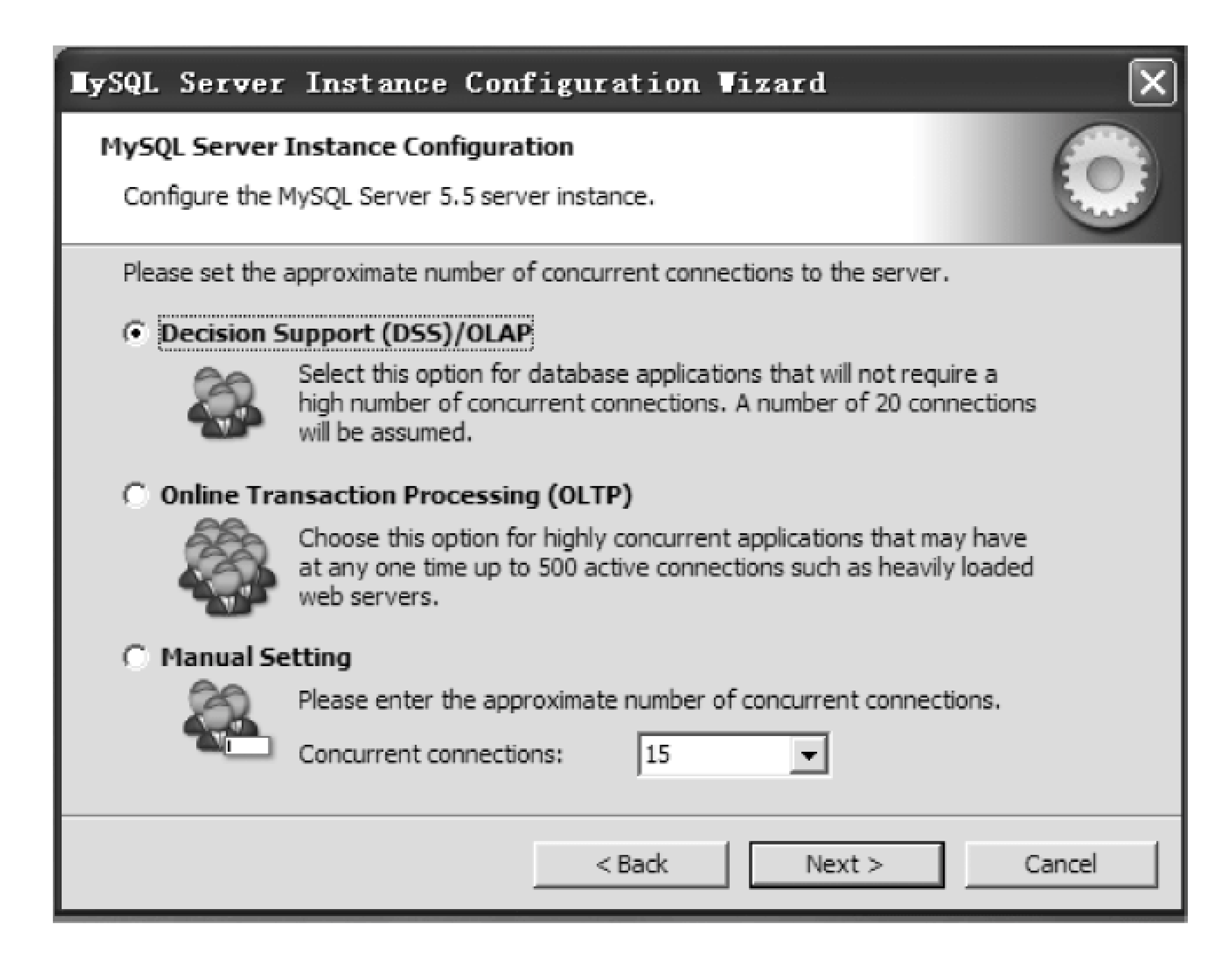


图 7.14 设置并发连接数界面

(6) 在并发连接界面中选择 Decision Support (DSS)/OLAP,然后单击 Next 按钮进入网络选项设置界面,如图 7.15 所示。默认情况是启用 TCP/IP 网络,默认端口为 3306。

(7) 在设置网络选项界面中选择 Enable TCP/IP Networking 和 Enable Strict Mode 复选框,然后单击 Next 按钮进入字符集设置界面,如图 7.16 所示,MySQL 提供了三种方式来设置字符集,它们分别为:Standard Character Set(标准字符集,MySQL 提供的标准字符集默认为 Latin1)、Best Support For Multilingualism(支持多国语言最好的字符集,默认值为 UTF8)、Manual Selected Default Character Set/Collation(手工设置字符集)。

(8) 在设置字符集选项界面中选中 Manual Selected Default Character Set/Collation 选项,在 Character Set 下拉框中将 latin1 修改为 gb2312。单击 Next 按钮进入 Windows 服务

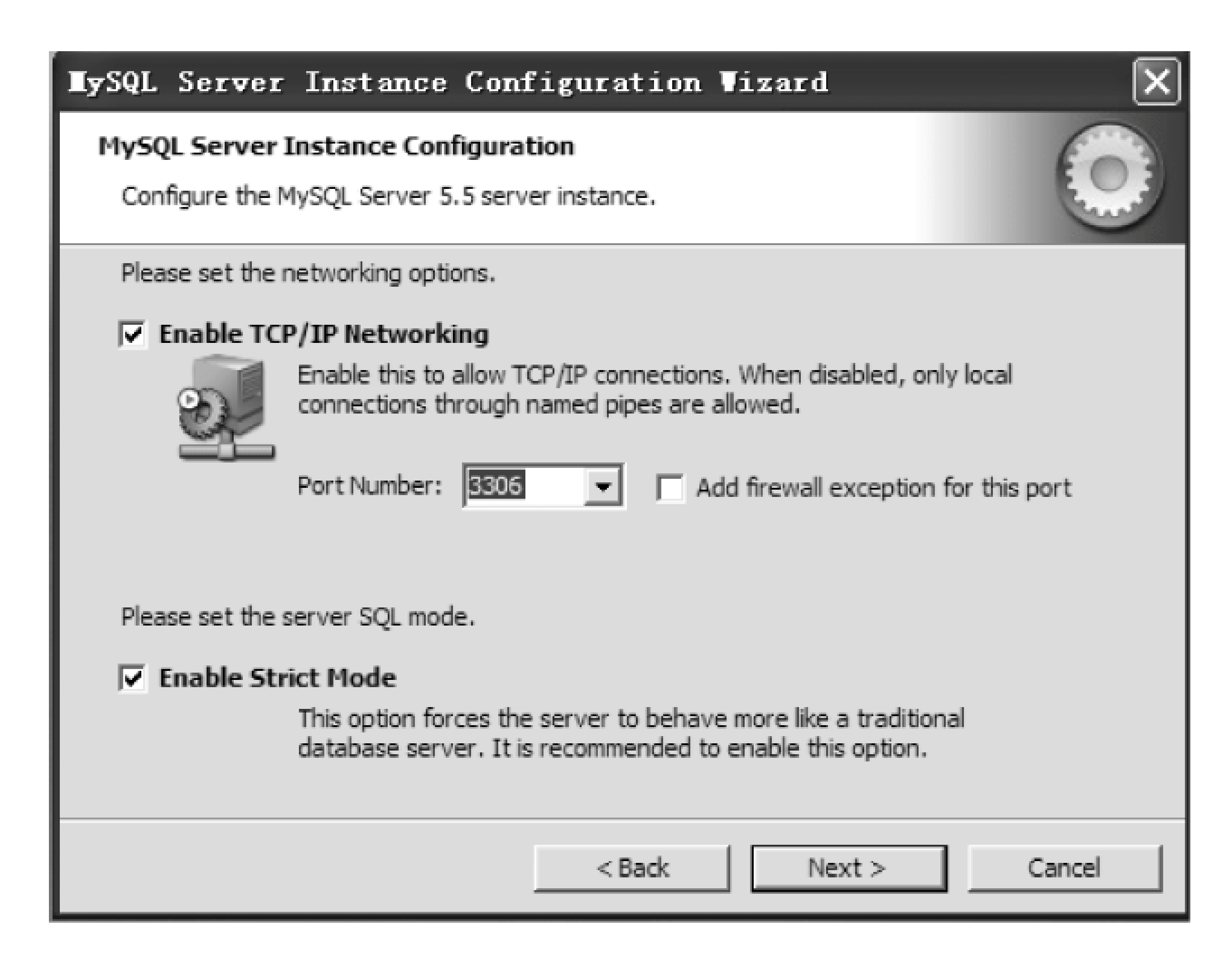


图 7.15　设置网络选项界面

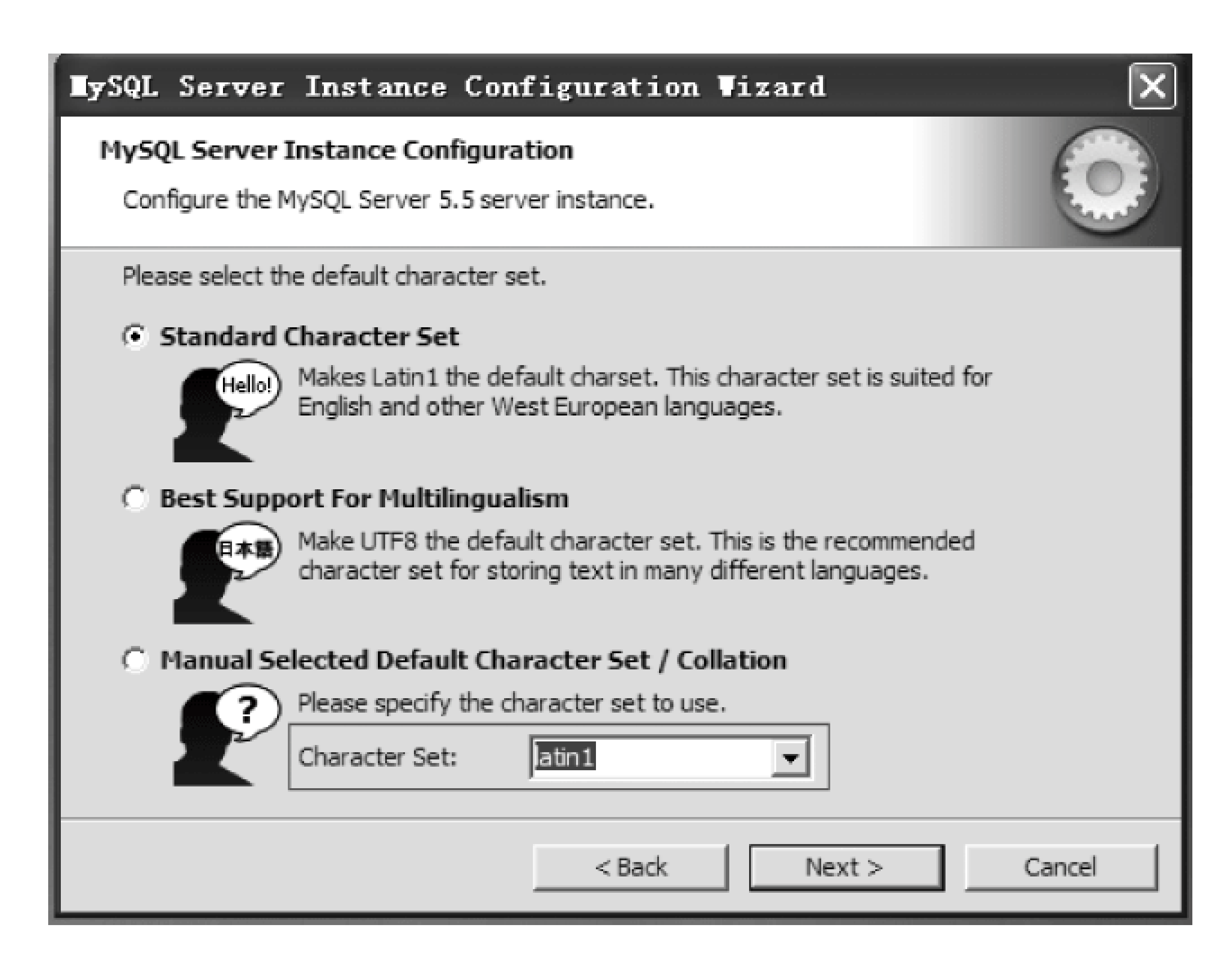


图 7.16　设置字符集选项界面

选项设置界面，如图 7.17 所示。

在 Windows 选项设置界面，选项 Install As Windows Service 设置 MySQL 是否将作为 Windows 的一个服务，而选项 Include Bin Directory in Windows PATH 设置 MySQL 的 Bin 目录是否写入 Windows 的环境变量中。

(9) 在 Windows 选项设置界面中将 MySQL 设置成名为 MySQL 的服务，且将数据库的 Bin 目录写入 Windows 的环境变量，然后单击 Next 按钮进入安全选项设置界面，如图 7.18 所示。

在设置安全选项界面中有两个安全复选框，Modify Security Settings 复选框确定是否修改用户的 root 密码，Create An Anonymous Accout 复选框确定是否创建一个匿名用户，在具体开发时，建议不要创建匿名用户。



图 7.17　设置 Windows 选项界面



图 7.18　设置安全选项界面

(10) 在安全选项设置界面中设置 root 用户的密码,此处密码设为"123456",然后单击 Next 按钮进入准备执行界面,如图 7.19 所示。

(11) 确认设置没有问题后,在准备执行界面中单击 Execute 按钮后开始执行。执行成功后的界面如图 7.20 所示。单击 Finish 按钮,结束 MySQL 的全部配置。

安装 MySQL 软件时,选择的是 Typical 类型,该类型下的 MySQL 软件的安装文件在 C:\Program Files\MySQL\MySQL Server 5.5 这个目录下,在该目录下有一个 my.ini 文件,这是 MySQL 的选项文件,MySQL 启动时会自动加载该文件中的一些选项。可以通过修改选项文件来修改 MySQL 的一些默认设置。MySQL 软件的数据库文件则安装在 C:\Documents and Settings\All Users\Application Data\MySQL\MySQL Server 5.5\data 目

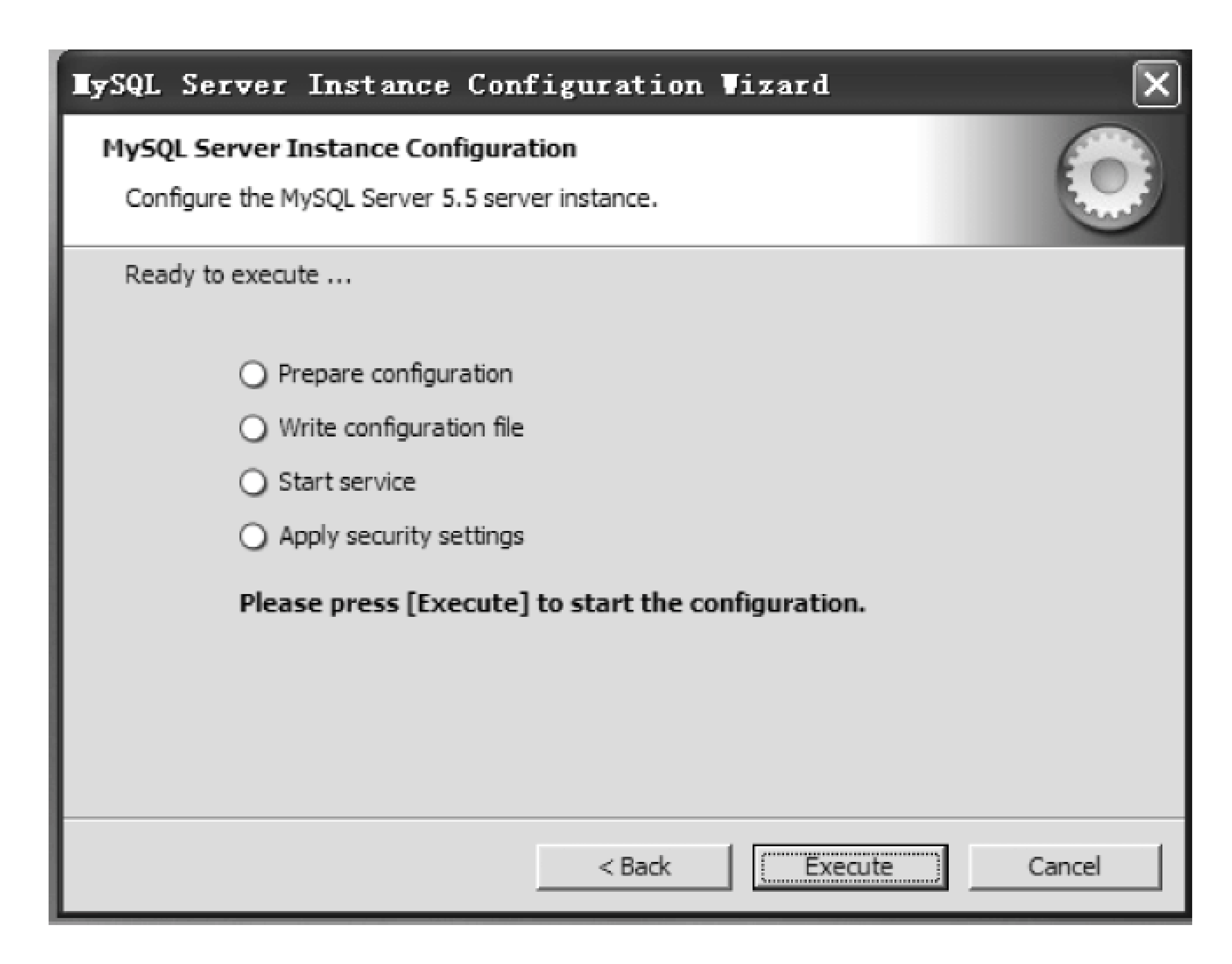


图 7.19 准备执行界面



图 7.20 安装完成界面

录下，在 data 目录中 MySQL 为每个数据库建立一个文件夹，所有的表文件存放在相应的数据库文件夹中。

7.3 MySQL 的卸载

实现卸载 MySQL 软件的具体步骤如下。

(1) 双击 MySQL 安装程序(mysql-5.5.38-win32.msi)，接着就会使用 Windows Installer 开始卸载过程，如图 7.21 所示。

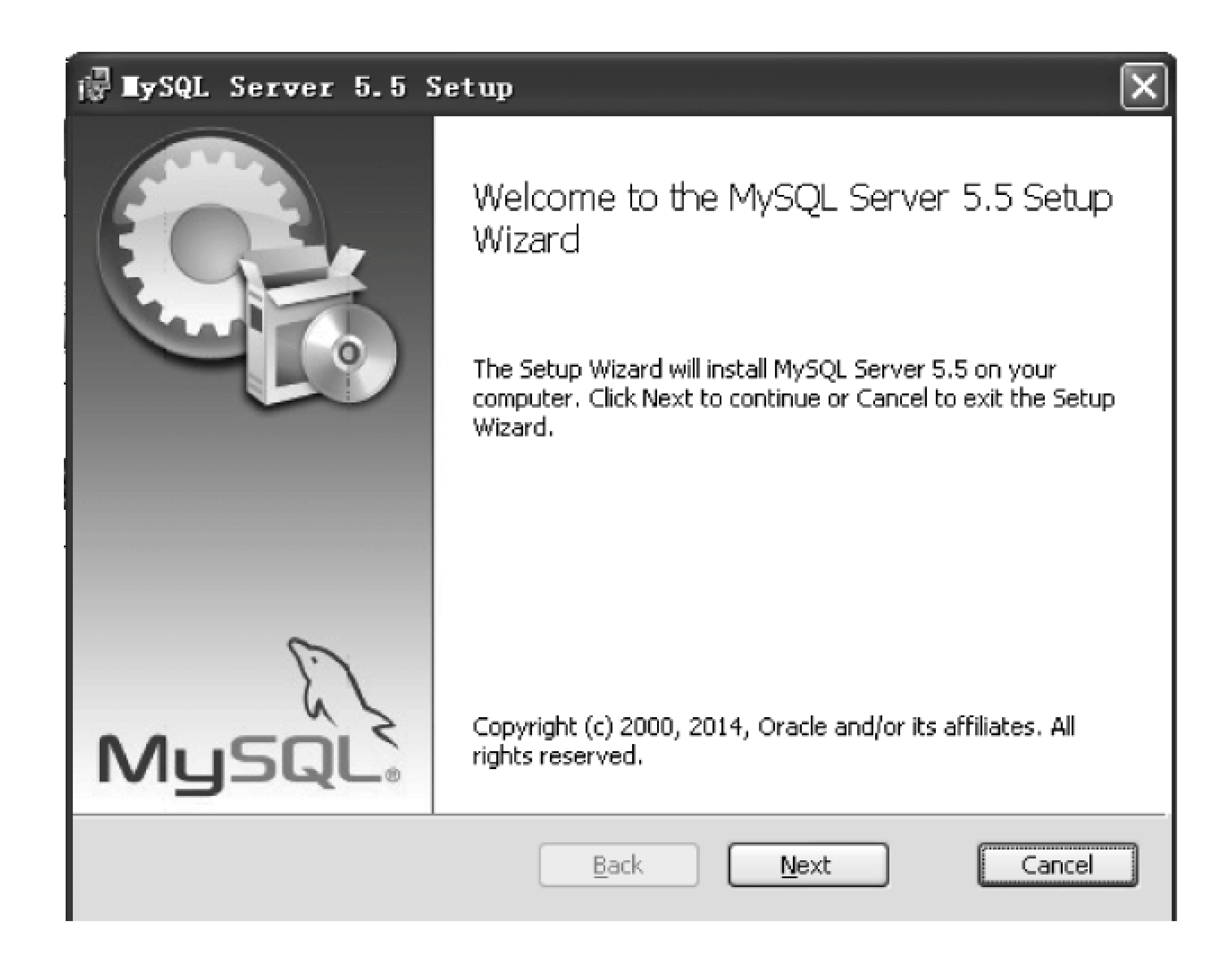


图 7.21　卸载欢迎界面

(2) 单击 Next 按钮后，在出现的对话框中单击 Remove 按钮进行卸载操作，如图 7.22 所示，就会进入 Ready to remove MySQL Server 5.5 对话框。



图 7.22　选择卸载 MySQL 操作界面

(3) 在 Ready to remove MySQL Server 5.5 对话框中确认一下下载的信息，单击 Remove 按钮开始对 MySQL 软件的卸载，如图 7.23 所示。卸载完成后就会出现确认卸载完成后的对话框，在该对话框中单击 Finish 按钮完成对 MySQL 软件的卸载。

上述步骤执行后，MySQL 软件就可以成功卸载。这时可以重新安装 MySQL 软件，注意：在重新安装之前，一定要把卸载 MySQL 软件时没有删除干净的 MySQL 文件删除掉，即需要删除路径为 C:\Program Files\MySQL 和 C:\Documents and Settings\All Users\Application Data\MySQL 的文件。



图 7.23 卸载信息确认页面

7.4 MySQL 软件的常用操作

7.3 节介绍了如何下载、安装、配置和卸载 MySQL，本节将详细介绍 MySQL 软件的常用操作：启动和关闭 MySQL 服务。

所谓 MySQL 服务是指一系列关于 MySQL 软件后台进程，只有启动了 MySQL 服务后，才可以连接 MySQL 软件进行操作。

1. 通过图形界面启动和关闭 MySQL 服务

对于 MySQL 数据库服务器，只有启动 MySQL 服务，客户端才能登录到 MySQL 数据库。在配置 MySQL 的过程中，已经设置 MySQL 服务作为 Windows 系统的服务，服务名为 MySQL。同时选择 Launch the MySQL Server automatically 复选框使 MySQL 服务为自动启动型。但由于每运行一个服务都会消耗一定的系统资源，因此对于一些经常不用的服务最好关闭，当需要使用时再开启。如何启动和关闭 MySQL 服务？具体步骤如下。

（1）右击"我的电脑"，在出现的菜单中选择"管理"命令，出现如图 7.24 所示的"计算机管理"窗口。

（2）在"计算机管理"窗口中选择"计算机管理(本地)"|"服务和应用程序"|"服务"结点，在右边窗口中会显示 Windows 系统的所有服务，其中包括名为 MySQL 的服务，如图 7.25 所示。

（3）查看 MySQL 服务可以发现服务已经处于启动状态，并且该服务的类型为自动类型。如果想修改 MySQL 服务的状态，可以单击"计算机管理"工具栏上的相应按钮。其中，按钮 ▶ 可以实现启动服务功能、按钮 ■ 可以实现停止服务功能、按钮 ■▶ 可以实现重新启动服务功能。



图 7.24 “计算机管理”窗口



图 7.25 MySQL 服务

(4) 对于 MySQL 服务,由于不是系统自带的服务,所以除了在需要时启动外,还需要设置启动类型为手动类型。在具体设置时,需右击 MySQL 服务,然后选择“属性”命令,出现“MySQL 的属性”对话框,如图 7.26 所示。在该对话框中不仅可以设置服务类型,而且还可以操作服务。

关于三种服务类型:自动,Windows 系统每次启动都会自动启动该类型服务,服务启动后可以手动将状态修改为停止、暂停和重新启动等;手动,该类服务需要手动启动,启动后可以改变服务状态,如停止、重新启动等;已禁用,该类服务不能启动,也不能改变服务状态。

对于 MySQL 服务,如果需要经常操作 MySQL 软件,那么该服务最好设置为自动启动类型;如果使用机会较少,可以将其设置为手动启动类型,这样可以避免 MySQL 服务长时间占用系统资源。



图 7.26　设置 MySQL 服务类型

2. 通过 DOS 窗口启动和关闭 MySQL 服务

除了可以通过图形界面来启动和关闭 MySQL 服务外,还可以通过 DOS 窗口启动和关闭 MySQL 服务。

具体步骤如下。

(1) 通过"开始"|"运行"命令打开"运行"对话框,然后在该对话框中输入"cmd"命令来打开 DOS 窗口,如图 7.27 所示。



图 7.27　"运行"对话框

(2) 在 DOS 窗口中,如果想查看 Windows 系统已经启动的服务,可以通过如下命令来实现:net start,具体运行过程如图 7.28 所示,可以发现 MySQL 软件的服务 MySQL 已经启动。

(3) 如果 MySQL 软件的服务 MySQL 已经启动,可以通过命令来关闭 MySQL 服务,具体命令如下:net　stop　MySQL,具体运行过程如图 7.29 所示。

(4) 如果 MySQL 软件的服务 MySQL 处于关闭状态,可以通过命令来启动 MySQL 服务,具体命令如下:net　start　MySQL,具体运行过程如图 7.30 所示。



图 7.28　查看已经启动的服务

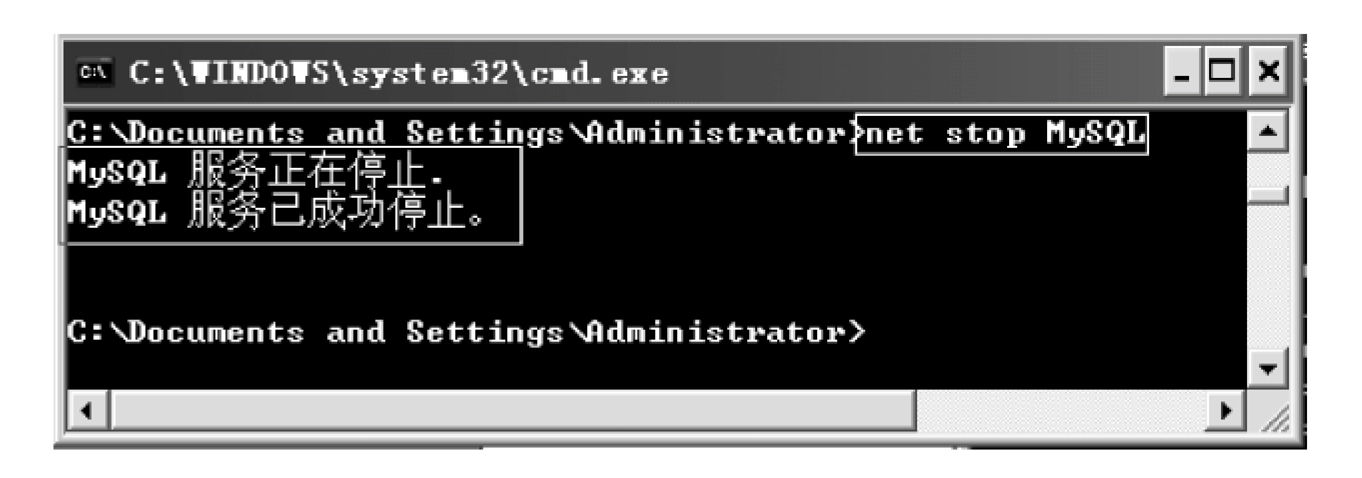


图 7.29　关闭 MySQL 服务



图 7.30　启动 MySQL 服务

7.5　MySQL 官方客户端软件

由于 MySQL 软件是基于 C/S 模式的数据库管理系统，因此可以通过各种客户端软件来与 MySQL 数据库管理系统相关联。针对 MySQL 软件，MySQL 公司开发了众多的客户端软件来帮助用户使用 MySQL 数据库管理系统，最著名的就是 MySQL Command Line Client 和 MySQL-Workbench 客户端软件。本节重点介绍 MySQL Command Line Client。

1. MySQL 命令行客户端软件

MySQL 安装和配置完后，一般都会安装一个简单命令行实用程序(MySQL Command Line Client)。该客户端没有下拉菜单，没有流行的用户界面，但它是众多 MySQL 客户端软件中实用功能最多的工具之一，它可以快速地登录和操作 MySQL 软件。

选择“开始”|“程序”|MySQL|MySQL Server 5.5|MySQL Command Line Client 菜单项，进入 MySQL 客户端，如图 7.31 所示。在客户端输入正确的密码，就以 root 用户身份登录到 MySQL 服务器。注意前提是 MySQL 软件的服务必须处于启动状态。



图 7.31　运行 MySQL Command Line Client 程序

当通过 MySQL Command Line Client 程序登录到服务器后，如图 7.32 所示，首先会输出一段欢迎内容和一个“MySQL＞”命令提示符。在该命令提示符后输入相应内容，比如 SQL 语句，同时并以分号(；)来结束，然后按 Enter 键就可以操作 MySQL 数据库。以 root 用户身份登录可以对数据库进行所有的操作。

```
MySQL 5.5 Command Line Client
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.37 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

图 7.32　登录到 MySQL 数据库

2. 通过 DOS 窗口连接 MySQL 软件

在 Windows 中还可以通过 DOS 窗口来登录 MySQL 软件，这种登录方式的具体步骤如下。

(1) 通过“开始”|“运行”命令打开“运行”对话框，然后在该对话框中输入“cmd”命令来打开 DOS 窗口。

(2) 在 DOS 窗口中，可以通过 mysql 命令来登录 MySQL 软件，具体命令如下：mysql-h 127.0.0.1-u root-p

代码说明：mysql 是登录 MySQL 软件的命令，选项-h 表示需要登录 MySQL 软件的 IP 地址，选项-u 表示登录 MySQL 软件的用户名，选项-p 表示登录 MySQL 软件的密码。

运行效果：具体运行过程如图 7.33 所示。当提示输入密码时，输入正确密码则会输出一段欢迎内容和一个“mysql>”提示符，这时表示登录到 MySQL 软件。

```
C:\WINDOWS\system32\cmd.exe - mysql -h 127.0.0.1 -u root -p
C:\Documents and Settings\Administrator>mysql -h 127.0.0.1 -u root -p
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.37 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

图 7.33　登录 MySQL 软件

通过上述步骤就可以成功连接到 MySQL 软件。当在 DOS 窗口中执行“mysql -h 127.0.0.1 -u root -p”命令时，有时会出现提示“mysql 不是内部命令，也不是可运行的程序和批处理文件”的错误，该错误原因是由于在配置 MySQL 软件时，在如图 7.34 所示的界面中没有选择 Include Bin Directory in Windows PATH 复选框。为了解决上述错误，除了可以重新配置 MySQL 软件外，还可以通过配置环境变量 path 来实现。具体步骤请查阅相关资料。



图 7.34　设置 MySQL 命令

7.6　MySQL 常用图形化管理软件 SQLyog 软件

除了 MySQL 官方提供的客户端软件外，很多公司也开发了自己的客户端软件。SQLyog 客户端软件由世界著名的 Webyog 公司研制，是专门针对 MySQL 数据库的图形

化管理工具。该客户端软件的突出特点是简洁高效、功能强大。

1．下载 SQLyog 软件

可以通过下面的步骤来实现该平台的下载，具体如下。

（1）首先访问下载 SQLyog 的官方网站（http://www.webyog.com/），如图 7.35 所示。

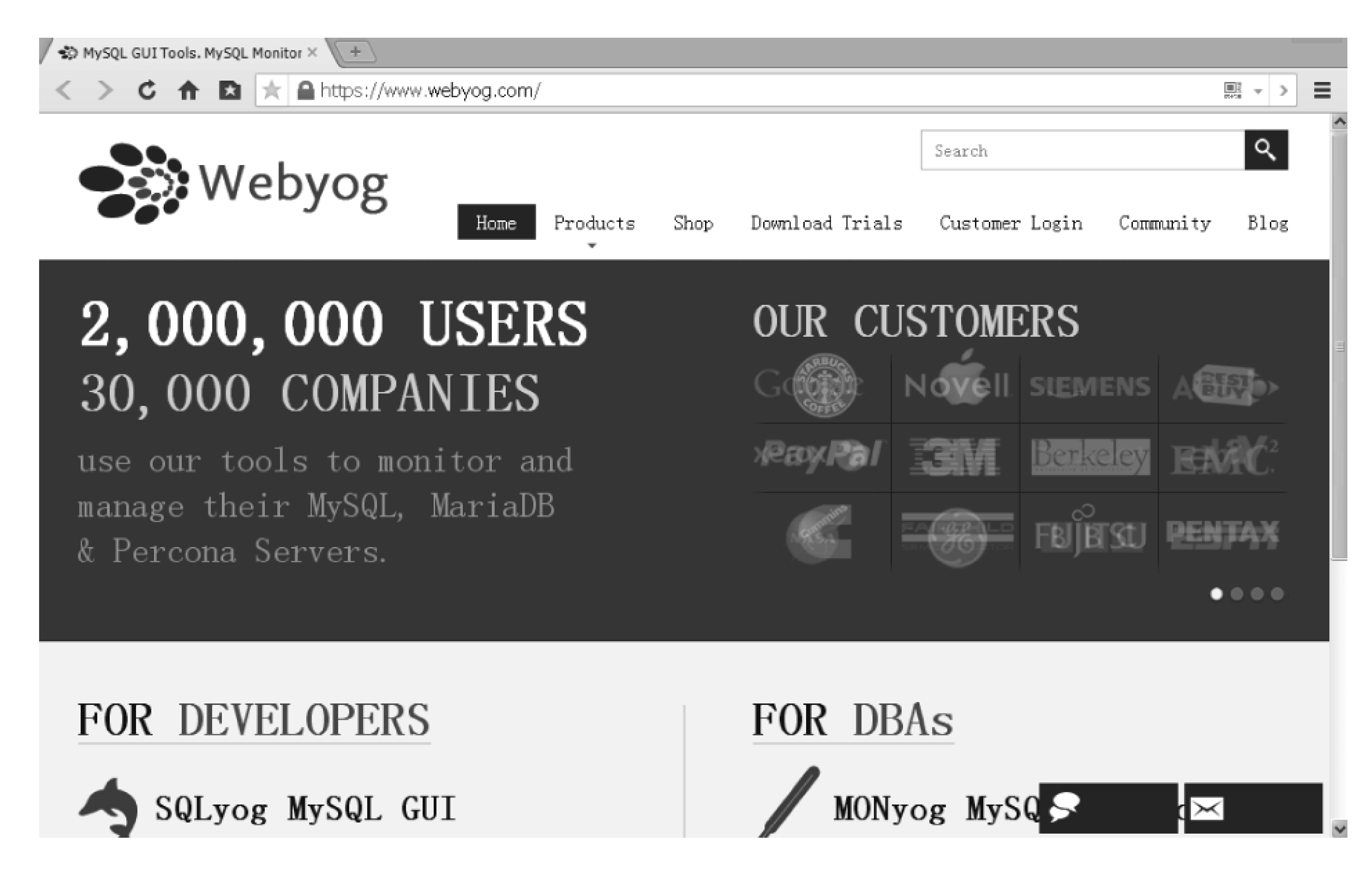


图 7.35　SQLyog 首页

（2）打开 SQLyog 首页后，单击 Downloads 超级链接，进入下载页面，如图 7.36 所示。在该页面中单击 Download-v11.5.0-1-32 bit 超级链接，就会实现 SQLyog 软件的下载。

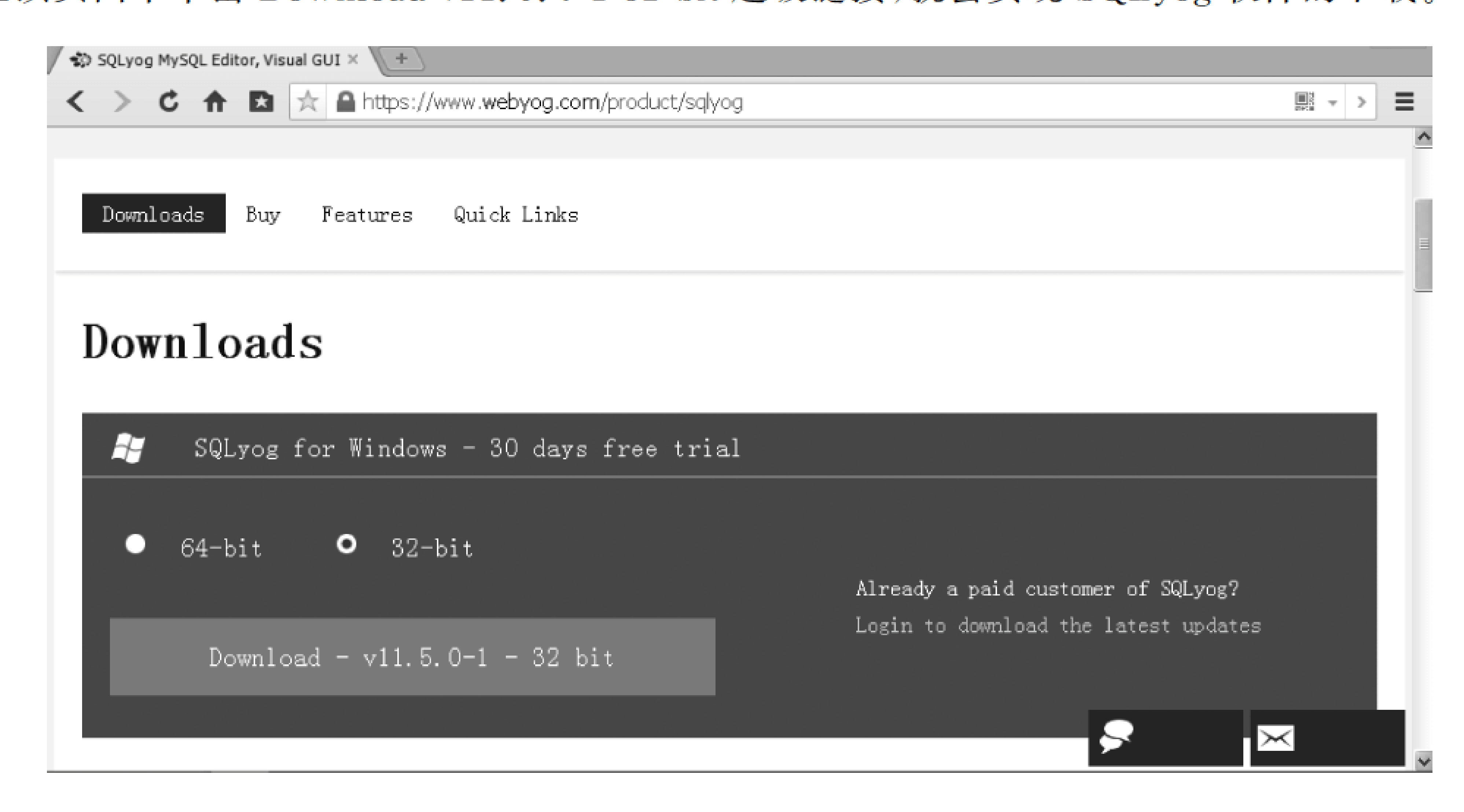


图 7.36　下载 SQLyog 客户端软件

SQLyog-11.5.0-1.x86Trial 为试用版,可以免费使用 30 天。

2. 安装 SQLyog 软件

下载完成后,就可以安装 SQLyog 客户端软件。具体安装步骤如下。

(1) 双击 SQLyog-11.5.0-1.x86Trial.exe 安装程序,使用 Windows Installer 开始安装过程,首先会出现 Please select a language 对话框,为了便于使用,在该对话框中选择 Chinese(Simplified)语句,单击 OK 按钮进入下一界面。

(2) 在欢迎界面中单击"下一步"按钮,进入"许可证协议"对话框,在对话框中选择"我接受许可证协议的条款"单选按钮,单击"下一步"按钮进入"选择组件"对话框。

(3) 在"选择组件"对话框中,选择相应的组件,然后单击"下一步"按钮进入"选择安装目录"对话框。

(4) 在"选择安装目录"对话框中,采用默认的安装目录设置,用户也可修改,然后单击"安装"按钮进行自动安装。当自动安装完成后,单击"下一步"按钮后,进入安装完成对话框,单击"完成"按钮完成对 SQLyog 客户端软件的安装。

3. 通过 SQLyog 客户端软件登录 MySQL 软件

SQLyog 客户端软件的功能非常全面,包含数据库的管理、数据对象的管理、数据导入和导出、数据管理等功能。

双击 SQLyog 快捷方式,打开该软件登录界面,正确输入服务器的地址、用户名、密码、端口(如图 7.37 所示),单击"连接"按钮就可以连接和登录数据库服务器了,这时会进入 SQLyog 软件的操作界面,如图 7.38 所示。

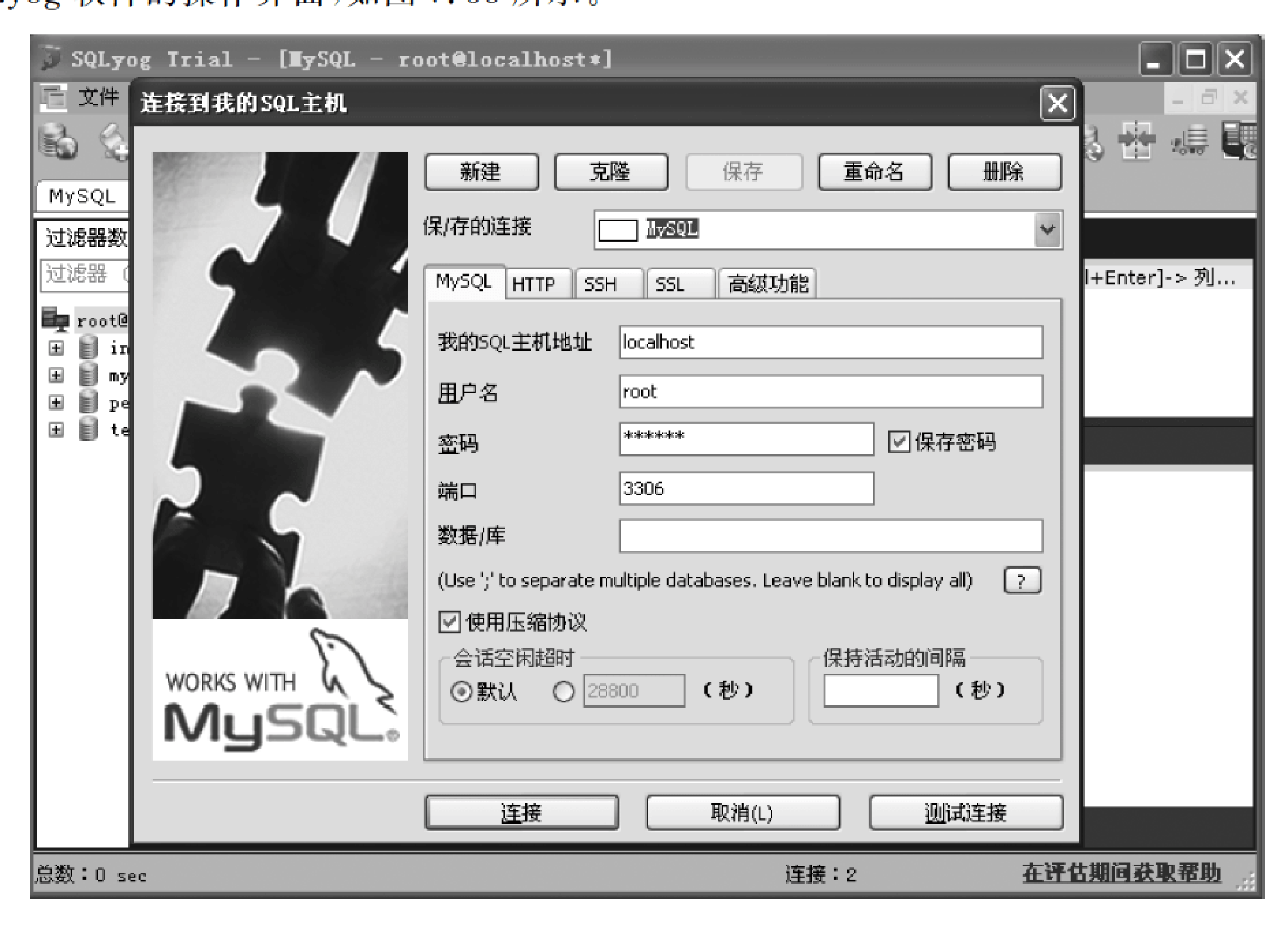


图 7.37 登录界面



图 7.38 操作界面

小结

本章介绍了目前常用的 MySQL 关系型数据库管理系统，包括 MySQL 软件的下载、安装以及卸载，MySQL 服务器的启动和停止，MySQL 的使用方法，包括命令行客户端以及界面客户端工具登录服务器。需重点掌握 MySQL 关系型数据库管理系统的使用（包括服务器的启动和关闭、登录服务器），为进一步学习 MySQL 软件应用打好基础。

习题

1. 安装 MySQL 服务器并进行配置。
2. 练习关闭、启动 MySQL 服务。
3. 练习使用命令行客户端登录服务器。

第8章 MySQL数据库的基本操作

本章要点：

- MySQL 数据库、数据库对象及数据类型；
- SQL 基础；
- MySQL 数据库备份与恢复。

数据库可以看成是存储数据对象的容器，这些数据对象包括表、视图等。其中，表是最基本的数据对象，用来存放数据。MySQL 通常采用命令行方式创建、操作数据库和数据对象，也可以通过界面管理工具创建、操作数据库和数据对象。数据库和表创建成功后，主要通过 SQL 语句来对表中数据进行操作，包括增加、修改、删除、查询数据。其中，数据的增加、修改、删除也可通过界面管理工具来完成。本章重点讨论 MySQL 数据库以及数据库对象等的创建与使用方法。

8.1 MySQL 数据库、数据库对象及数据类型

对于 MySQL 服务器，当连接上 MySQL 服务器后，就可以操作数据库里存储到数据库对象中的数据了。

8.1.1 MySQL 数据库分类

在 MySQL 软件中，数据库可以分为系统数据库和用户数据库两大类。

1. 系统数据库

系统数据库是指安装完 MySQL 服务器后，会附带的一些数据库，如图 8.1 所示。系统数据库会记录一些必需的信息，用户不能直接修改这些系统数据库。各个系统数据库的作用如下。

（1）information_schema：主要存储了系统中的一些数据库对象信息，例如用户表信息、列信息、权限信息、字符集信息等。

（2）performance_ schema：主要存储数据库服务器性能参数。

（3）mysql：主要存储了系统的用户权限信息。

图 8.1 系统数据库

（4）test：该数据库为 MySQL 数据库管理系统自动创

建的测试数据库,任何用户都可以使用。

2. 用户数据库

用户数据库是用户根据实际需求创建的数据库,如图 8.2 所示。其中,xscj 数据库就属于用户数据库。数据库可以看作是一个存储数据对象的容器,在 MySQL 中,这些数据对象主要包含表、视图、存储过程、函数、触发器和事件等。

在 SQLyog 客户端软件的"对象资源管理器"中,每个数据库结点下都拥有一个树状路径结构,如图 8.3 所示。其中,树状结构中的每个具体子结点都是数据库对象。

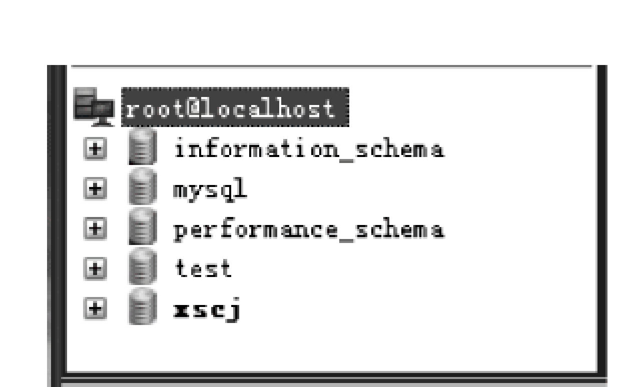


图 8.2 用户数据库



图 8.3 数据库对象

8.1.2 MySQL 的数据库对象

数据库可以看作是一个存储数据对象的容器,在 MySQL 中,这些数据对象包括以下几种。

1. 表

表是 MySQL 中最主要的数据库对象,是用来存储和操作数据的一种逻辑结构。表由行和列组成,因此也称为二维表。

2. 视图

视图是从一个或多个基本表中引出的表。本质上是一种虚拟表,其内容与真实的表相似,包含一系列带有名称的列和行数据,在数据库中只存放视图的定义,而不存放视图对应的数据,这些数据仍存放在导出视图的基本表中。当基本表的数据发生变化时,从视图中查询出来的数据也随之改变。视图一经定义,就可以像基本表一样被查询、修改、删除和更新。

3. 存储过程和函数

存储过程和函数用来实现将一组关于表的操作的 SQL 语句代码当作一个整体来执行,两者的区别主要是函数必须有返回值,而存储过程则没有。在数据库系统中,当调用存储过程和函数时,则会执行这些对象中所设置的 SQL 语句组从而实现相应的功能。

4. 触发器和事件

触发器用来实现由一些表事件(例如表中数据更新或删除时)触发某个操作。在数据库系统中,当执行表事件时,则会激活触发器从而执行其包含的操作。

8.1.3 MySQL 数据类型

在创建表的列时,必须为其指定数据类型,列的数据类型决定了数据的取值、范围和存储格式。MySQL 提供了丰富的数据类型,如表 8.1 所示。

表 8.1 MySQL 数据类型

数据类型	符号标志
整数型	BIGINT, INT, SMALLINT, MEDIUMINT, TINYINT
精确数值型	DECIMAL, NUMERIC
浮点型	FLOAT, REAL, DOUBLE
位型	BIT
字符型	CHAR, VARCHAR, LONGVARCHAR, LONGTEXT
Unicode 字符型	NCHAR, NVARCHAR
BLOB 类型	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
文本型	TEXT, TINYTEXT
二进制型	BINARY, VARBINARY
日期时间类型	DATE, TIME, DATETIME, TIMESTAMP, YEAR

8.2 SQL 基础

关系型数据库的标准语言是 SQL(Structured Query Language,结构化查询语言)。SQL 是关系数据库操作的基础语言,将数据查询、数据操纵、数据定义、事务控制、系统控制等功能集于一体,使得数据库应用开发人员、数据库管理员等都可以通过 SQL 实现对数据库的访问和操作。

SQL 之所以成为关系数据库的标准语言,并得到广泛的应用,其原因在于 SQL 语句具有以下特点。

(1) 功能一体化。几乎涵盖了对数据库的所有操作,语言风格统一。

(2) 高度的非过程化。在使用 SQL 操作数据库时,用户只需要说明"做什么",而不需要说明"怎么做"。用户任务的实现对用户而言是透明的,由系统自动完成。这大大减轻了用户的负担,同时降低了对用户的技术要求。

(3) 面向集合的操作方式。SQL 采用集合操作方式,不仅查询结果可以是多条记录的集合,而且一次插入、删除、修改操作的对象也可以是多条记录的集合。面向集合的操作方式提高了对数据库的操作效率。

(4) 多种使用方式。SQL 既是自含式语言,又是嵌入式语言。作为自含式语言,它能够独立地用于联机交互的使用方式,用户可以在终端键盘上直接输入 SQL 命令对数据库进行

操作；作为嵌入式语言，SQL 语句能够嵌入到高级语言（例如 C、C++、Java 等）程序中，供程序员设计程序时使用。

（5）简洁、易学。SQL 命令数量有限，语法简单，接近于自然语言（英语），因此容易学习和掌握。

根据实现功能的不同，SQL 可以分为以下 4 类。

（1）数据定义语言：用于定义、修改、删除数据库对象，包括 CREATE、ALTER、DROP。

（2）数据操作语言：用于改变数据库中的数据，包括数据插入（INSERT）、数据修改（UPDATE）和数据删除（DELETE）。

（3）数据查询语言：用于数据检索，包括 SELECT。

（4）数据控制语言：用于对数据库对象进行访问权限的控制，包括 GRANT、REVOKE、AUDIT 和 NOAUDIT 等。

8.2.1 数据定义语句和数据操作语句

1. 命令行方式创建数据库和表

1）MySQL 数据库相关操作

（1）创建数据库。在 MySQL 中创建数据库通过 SQL 语句 CREATE DATABASE 来实现。语法格式：

```
CREATE  Database  database_name;
```

【例 8.1】 创建一个名为 xscj 的数据库。语句为：

```
mysql > create database xscj;
```

运行效果：执行上面的 SQL 语句，其结果如图 8.4 所示。

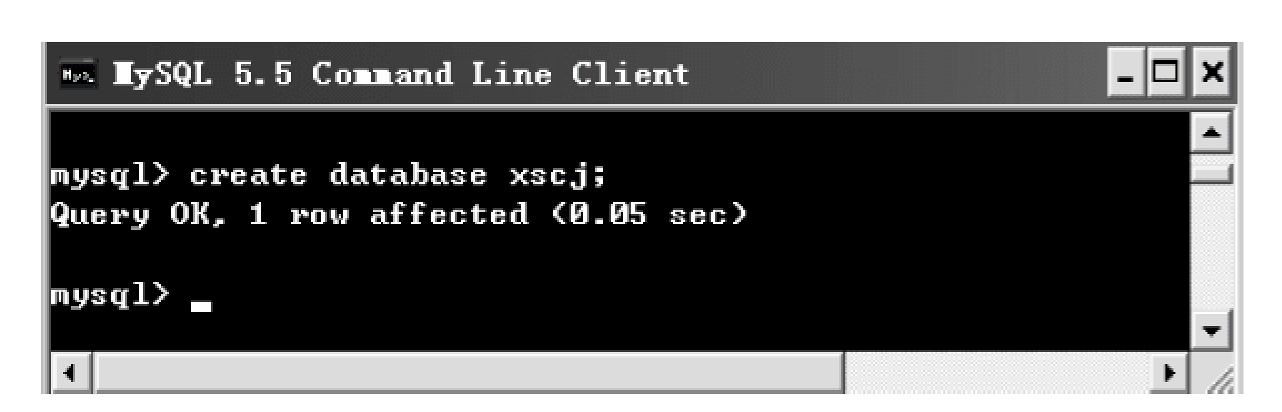


图 8.4　创建数据库

注意：在 MySQL 中，每一条 SQL 语句都以“;”作为结束标志。该标点符号为英文标点符号。

（2）查看数据库。在 MySQL 中查看已经存在的数据库通过 SQL 语句 SHOW DATABASES 来实现。语法格式：

```
SHOW  DATABASES;
```

执行上面的语句，会显示 MySQL 软件中所有的数据库，包括存在的用户数据库和 4 个系统数据库，且以列表形式显示，其结果如图 8.5 所示。

（3）选择数据库。数据库是数据库对象的容器，而在数据库管理系统中一般会存在多个

数据库,在操作数据库对象之前,首先要确定是哪一个数据库。即在对数据库对象进行操作时,需要选择一个数据库。在 MySQL 中选择数据库通过 SQL 语句 USE 来实现。语法格式:

```
USE  database_name;
```

【例 8.2】 执行 SQL 语句 USE,选择名为 xscj 的数据库。

```
mysql> USE  xscj;
```

执行上面的语句,其结果如图 8.6 所示。

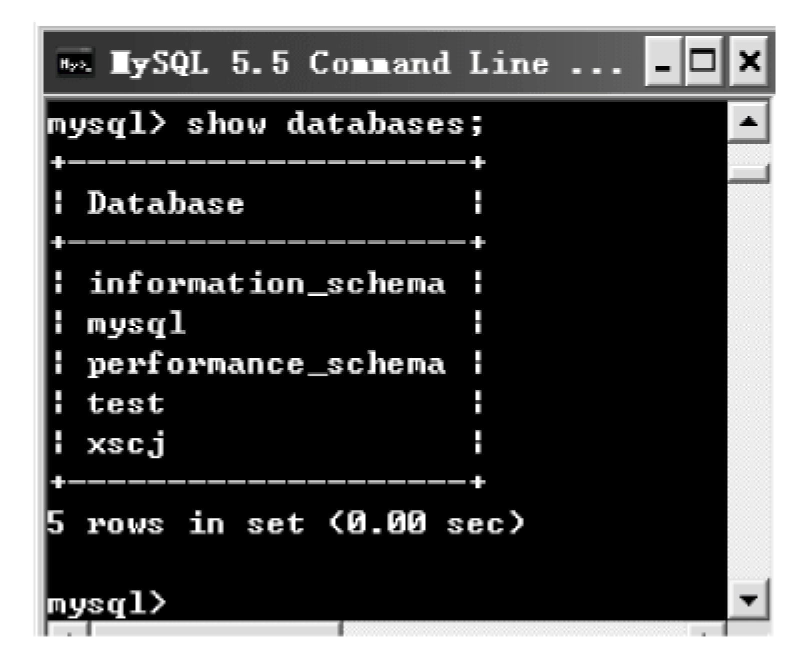


图 8.5　显示所有数据库



图 8.6　选择数据库

(4) 删除数据库。已经创建的数据库需要删除,在 MySQL 中使用 DROP DATABASE 来实现。语法格式:

```
DROP DATABASE database_name;
```

【例 8.3】 执行 SQL 语句删除数据库,首先创建一个名为 xscj1 的数据库,然后再删除该数据库。具体步骤如下。

```
mysql> create database xscj1;
mysql> show databases;
mysql> drop database xscj1;
mysql> show databases;
```

运行效果如图 8.7 和图 8.8 所示。

```
mysql> create database xscj1;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
| xscj               |
| xscj1              |
+--------------------+
6 rows in set (0.00 sec)

mysql> _
```

图 8.7　创建数据库

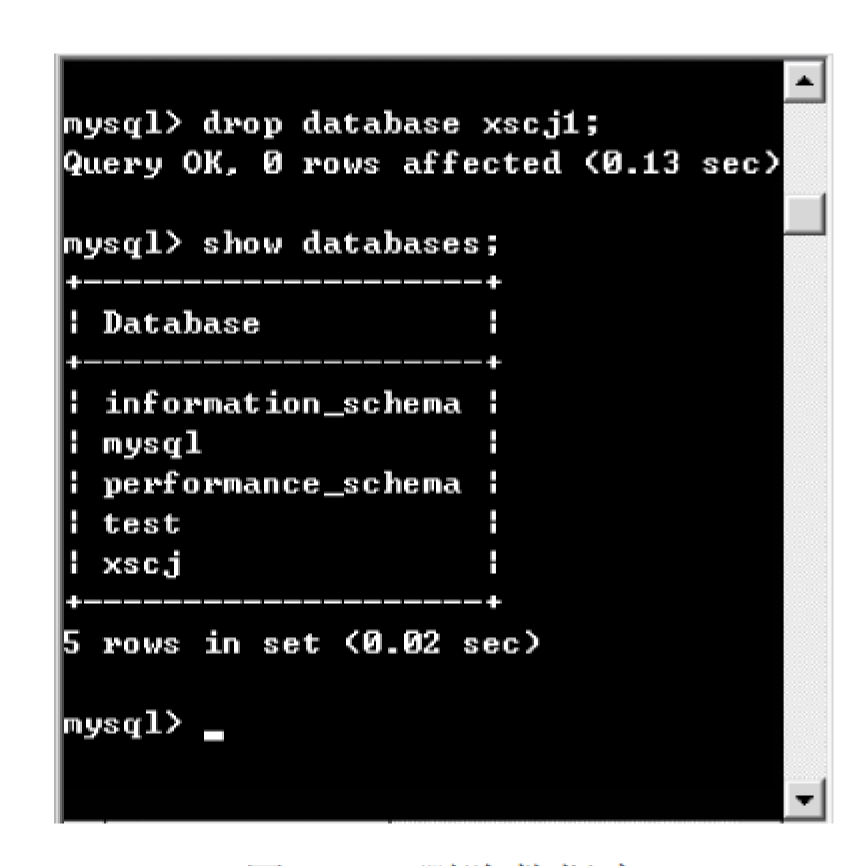


图 8.8　删除数据库

注意：这个命令必须小心使用，因为它将删除指定的整个数据库，该数据库的所有表（包括其中的数据）也将永久删除。

2）MySQL 数据表的操作

表的操作包括创建表、查看表、删除表和修改表。本节详细介绍表的操作。

（1）创建表。在 MySQL 数据库管理系统中创建表使用 SQL 语句 CREATE TABLE，基本格式如下：

```
CREATE TABLE table_name
(
    <列名 1> <数据类型> [<列选项>],
    <列名 2> <数据类型> [<列选项>],
    …
    <表选项>
)
```

上述语句中，table_name 表示要创建的表名字，表的具体内容定义在圆括号之中，各列之间用逗号分隔。“列名”指定表中字段的名称，“数据类型”指定字段的数据类型。

【例 8.4】 在 xscj 数据库中创建名为 xs 的表。

```
mysql > CREATE TABLE xs(
   sno varchar(6) NOT NULL,
   sname varchar(10) NOT NULL,
   sex tinyint NOT NULL,
   bir date,
   dno varchar(6));
```

运行效果如图 8.9 所示。

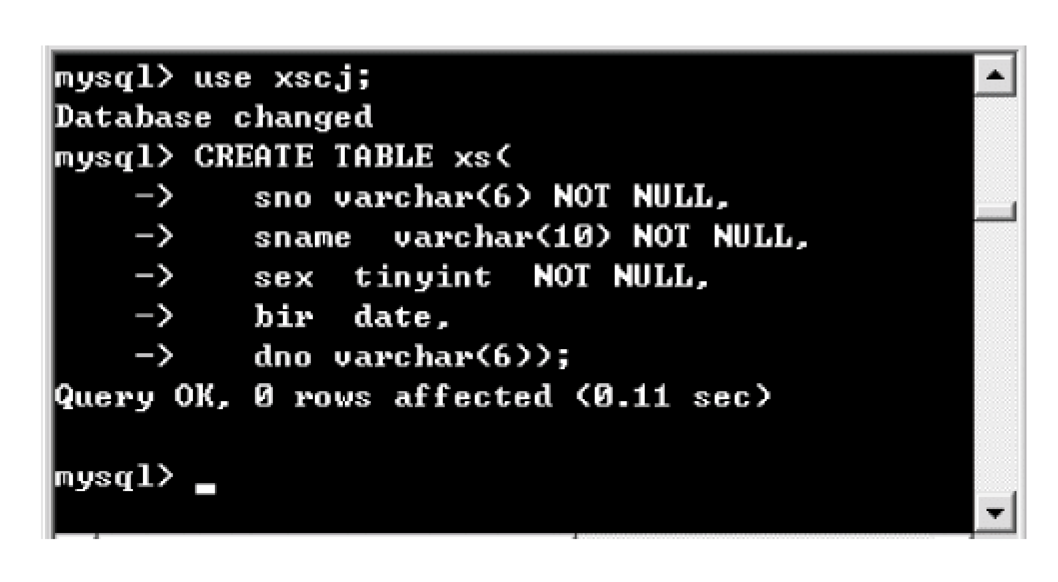


图 8.9 创建 xs 表

注意：创建表之前，需要选择数据库，如果没有选择数据库，创建表时会出现“No database selected”错误。

（2）查看表结构。当创建完表后，经常要查看表信息，如查看一下表的定义，通过执行 SQL 语句 DESCRIBE 来实现。语法格式：

```
DESCRIBE   table_name;
```

【例 8.5】 查看数据库 xscj 中 xs 表的结构信息。

具体步骤：

```
mysql > use xscj;
mysql > describe xs;
```

执行上面的 SQL 语句,运行效果如图 8.10 所示。

```
mysql> use xscj;
Database changed
mysql> describe xs;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| sno   | varchar(6)  | NO   |     | NULL    |       |
| sname | varchar(10) | NO   |     | NULL    |       |
| sex   | tinyint(4)  | NO   |     | NULL    |       |
| bir   | date        | YES  |     | NULL    |       |
| dno   | varchar(6)  | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
5 rows in set (0.05 sec)

mysql>
```

图 8.10　查看表定义信息

(3) 修改表。在 MySQL 语句中通过 ALTER TABLE 更改原有的表结构。例如,可以增加或删减列,创建或取消索引,更改原有列的类型,重新命名列或表,还可以更改表的描述和表的类型。

ALTER TABLE 语句的基本格式如下:

```
ALTER TABLE table_name
ADD (<列名> <数据类型> <列选项>)                          /* 添加列 */
| ALTER (<列名> {SET DEFAULT default_value | DROP DEFAULT})  /* 修改默认值 */
| CHANGE (<旧列名> <新列名> <数据类型> <列选项>)  /* 对列重命名 */
| MODIFY (<列名> <数据类型> <列选项>)               /* 修改列类型 */
| DROP <列名>)                                      /* 删除列 */
| RENAME <新表名>                                   /* 重命名该表 */
| 其他
```

【例 8.6】 为 xs 表添加总学分和备注两列,语句为:

```
ALTER TABLE xs ADD (zxf int default 0 ,bz text  NULL);
```

执行效果如图 8.11 所示。查看表结构,新增加的属性列已出现在表中,如图 8.12 所示。

```
mysql> ALTER TABLE xs  ADD  (zxf int default 0 ,bz text  NULL);
Query OK, 0 rows affected (0.20 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

图 8.11　xs 表增加属性列

更改表结构,将 xs 表中 sex 字段的字段类型改为 CHAR(2)。

```
ALTER TABLE xs CHANGE sex sex CHAR(2);
```

(4) 删除表。在 MySQL 数据库管理系统中删除表通过 SQL 语句 DROP TABLE 来实现。语法格式:

```
mysql> describe xs;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| sno   | varchar(6)  | NO   |     | NULL    |       |
| sname | varchar(10) | NO   |     | NULL    |       |
| sex   | tinyint(4)  | NO   |     | NULL    |       |
| bir   | date        | YES  |     | NULL    |       |
| dno   | varchar(6)  | YES  |     | NULL    |       |
| zxf   | int(11)     | YES  |     | 0       |       |
| bz    | text        | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)

mysql>
```

图 8.12　查看表定义

```
DROP TABLE  table_name
```

【例 8.7】 删除 xscj 数据库中的 xs 表，具体步骤如下。

```
mysql > use xscj;                                //选择 xscj 数据库
mysql > DROP TABLE xs;
```

执行效果如图 8.13 所示。

```
mysql> use xscj;
Database changed
mysql> drop table xs;
Query OK, 0 rows affected (0.03 sec)

mysql> _
```

图 8.13　删除表

为了检验数据库 xscj 中 xs 表是否还存在，执行 describe xs 语句，运行效果如图 8.14 所示。执行结果显示，xs 表已经不存在，表示删除 xs 表成功。

```
mysql> describe xs;
ERROR 1146 (42S02): Table 'xscj.xs' doesn't exist
mysql>
```

图 8.14　查看表

2. 命令行方式进行数据操作

(1) 插入记录。通过 INSERT 语句可以向表中插入一行或多行数据。

基本格式如下：

```
INSERT [INTO] table_name |view_name[(column_name1[,column_name2… ])]
VALUES (value1[,values, … ]);
```

【例 8.8】 向 xscj 数据库的表 xs(表中列包括学号、姓名、性别、出生日期、所在系编号)中插入如下一行：130601，赵林林，1，1995-02-10，070001。具体的 SQL 语句如下。

```
INSERT INTO XS VALUES('130601','赵林林',1,CAST('1995-02-10'as date),'070001');
```

说明：CAST(exp as type)为类型转换函数，参数 exp 是表达式，参数 type 是要转换的类型。本实例使用该函数将字符串'1995-02-10'转换成日期类型。

运行效果如图 8.15 所示。

```
MySQL 5.5 Command Line Client
mysql> INSERT INTO xs
    -> VALUES('130601','赵林林',1,CAST('1995-02-10' as date),'070001');
Query OK, 1 row affected (0.03 sec)

mysql>
```

图 8.15　命令行插入数据记录

为了校验插入是否成功，可以通过 SQL 语句 SELECT 来实现。具体 SQL 语句如下：

```
mysql > SELECT * FROM xs;
```

执行上面的 SQL 语句，其结果如图 8.16 所示。

```
MySQL 5.5 Command Line Client
mysql> select * from xs;
+--------+--------+-----+------------+--------+
| sno    | sname  | sex | bir        | dno    |
+--------+--------+-----+------------+--------+
| 130601 | 赵林林 |   1 | 1995-02-10 | 070001 |
+--------+--------+-----+------------+--------+
1 row in set (0.00 sec)

mysql>
```

图 8.16　查看表数据记录

(2) 修改记录。向表中插入数据后，如要修改表中的数据，可以使用 UPDATE 语句，基本格式如下：

```
UPDATE table_name
SET column_name1 = expr1 [, column_name2 = expr2 … ]
[WHERE condition]
```

【例 8.9】 将 xscj 数据库 xs 表中学号为“130601”学生的姓名改为“赵林”。具体的 SQL 语句如下：

```
UPDATE xs SET 姓名 = '赵林' WHERE 学号 = '130601';
```

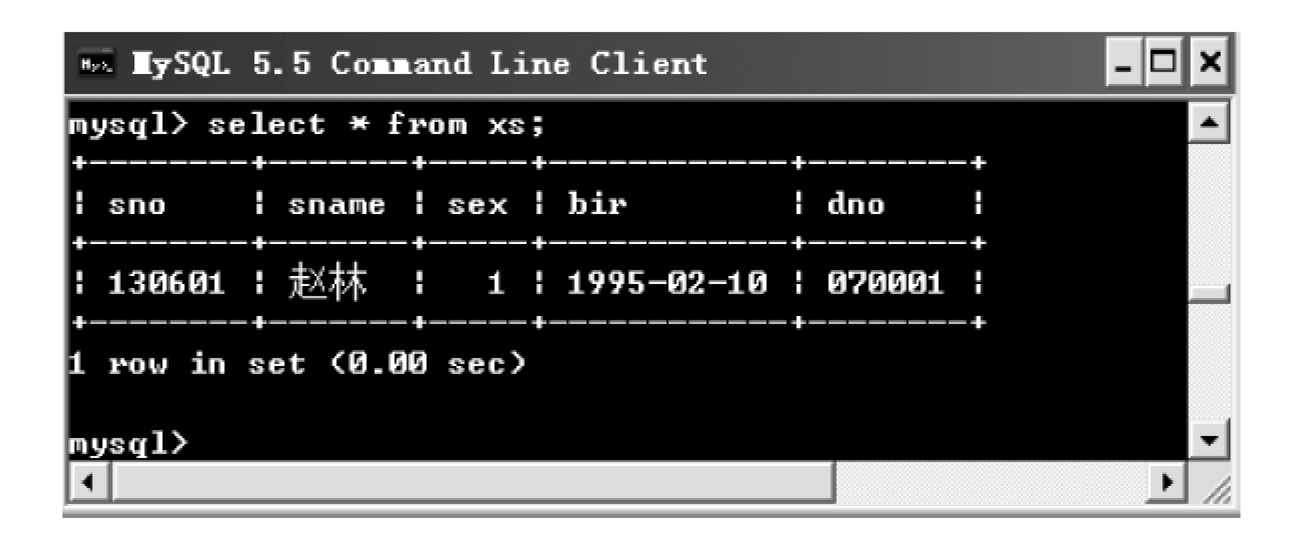
为了校验修改是否成功，可以通过 SQL 语句 SELECT 来实现。具体 SQL 语句如下：

```
SELECT * FROM xs;
```

执行上面的 SQL 语句，其结果如图 8.17 所示。

(3) 删除记录。删除表中数据一般使用 DELETE 语句，语法格式如下：

```
DELETE FROM table_name [WHERE condition]
```



```
MySQL 5.5 Command Line Client
mysql> select * from xs;
+--------+-------+-----+------------+--------+
| sno    | sname | sex | bir        | dno    |
+--------+-------+-----+------------+--------+
| 130601 | 赵林  |   1 | 1995-02-10 | 070001 |
+--------+-------+-----+------------+--------+
1 row in set (0.00 sec)

mysql>
```

图 8.17 查看表数据记录

【例 8.10】 要删除李芳的信息可使用如下语句：

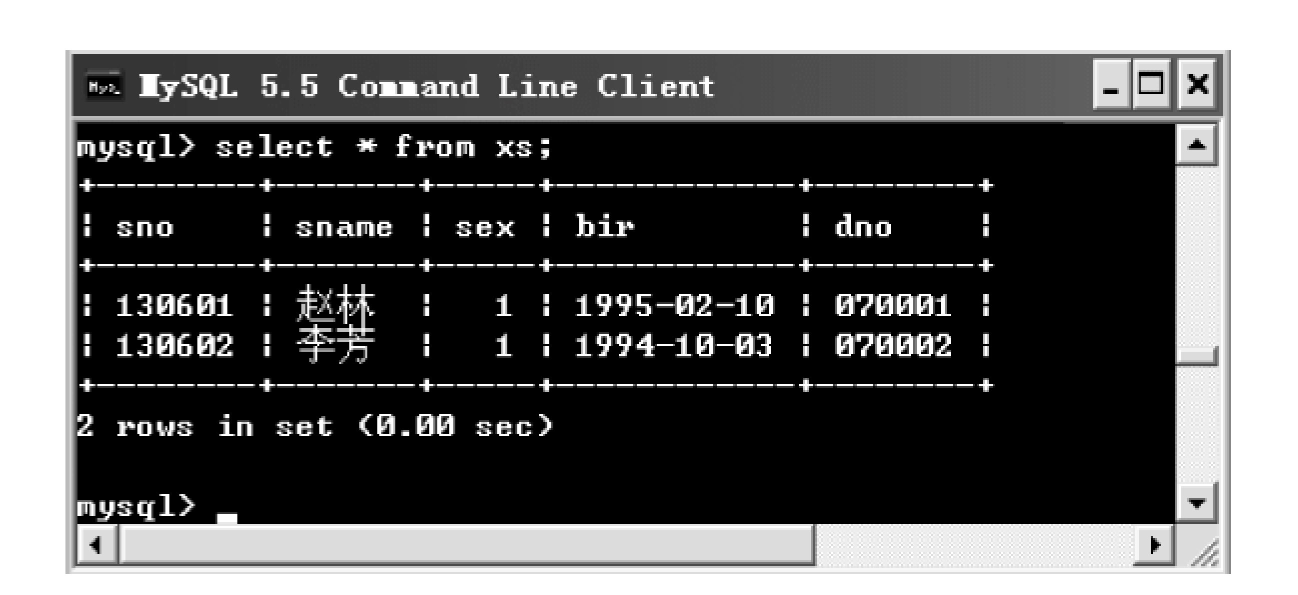
```
DELETE FROM XSWHERE sname = '李芳';
```

为了校验删除是否成功，可以通过 SQL 语句 SELECT 来实现。具体的 SQL 语句如下：

```
mysql > INSERT INTO xs VALUES('130602','李芳',1,CAST('1994 - 10 - 3' as date),'070001');
mysql > SELECT * FROM xs;
```

执行上面的 SQL 语句，其结果如图 8.18 所示。

```
mysql > DELETE FROM xs WHERE sname = '李芳';
```



```
MySQL 5.5 Command Line Client
mysql> select * from xs;
+--------+-------+-----+------------+--------+
| sno    | sname | sex | bir        | dno    |
+--------+-------+-----+------------+--------+
| 130601 | 赵林  |   1 | 1995-02-10 | 070001 |
| 130602 | 李芳  |   1 | 1994-10-03 | 070002 |
+--------+-------+-----+------------+--------+
2 rows in set (0.00 sec)

mysql> _
```

图 8.18 删除操作前的数据记录

运行效果如图 8.19 所示。

```
MySQL 5.5 Command Line Client
mysql> DELETE FROM xs WHERE sname='李芳';
Query OK, 1 row affected (0.02 sec)

mysql> select * from xs;
+--------+-------+-----+------------+--------+
| sno    | sname | sex | bir        | dno    |
+--------+-------+-----+------------+--------+
| 130601 | 赵林  |   1 | 1995-02-10 | 070001 |
+--------+-------+-----+------------+--------+
1 row in set (0.00 sec)
```

图 8.19 执行删除操作

3. MySQL 界面工具创建数据库和表

通过 MySQL 数据库服务器自带的 MySQL Command Line Client 来创建数据库,需要掌握 SQL 语句。在具体实践中,用户可以通过客户端软件 SQLyog 来创建数据库。

【例 8.11】 创建一个名为 xscj 的数据库。

(1) 首先连接数据库服务器,在“对象资源管理器”窗口中显示 MySQL 数据库管理系统中的所有数据库,如图 8.20 所示。



图 8.20 连接 MySQL 数据库管理系统

(2) 右击“对象资源管理器”窗口空白处,在出现的菜单中选择“新建数据库”命令,如图 8.21 所示,则会打开“创建数据库”对话框。



图 8.21 创建数据库命令

(3) 在出现的“创建数据库”对话框中,设置数据库名称为 xscj,然后单击“创建”按钮则会创建数据库,如图 8.22 所示。

(4) 当数据库 xscj 创建成功后,“对象资源管理器”中就会显示出名为 xscj 的数据库,如图 8.23 所示。



图 8.22　创建数据库

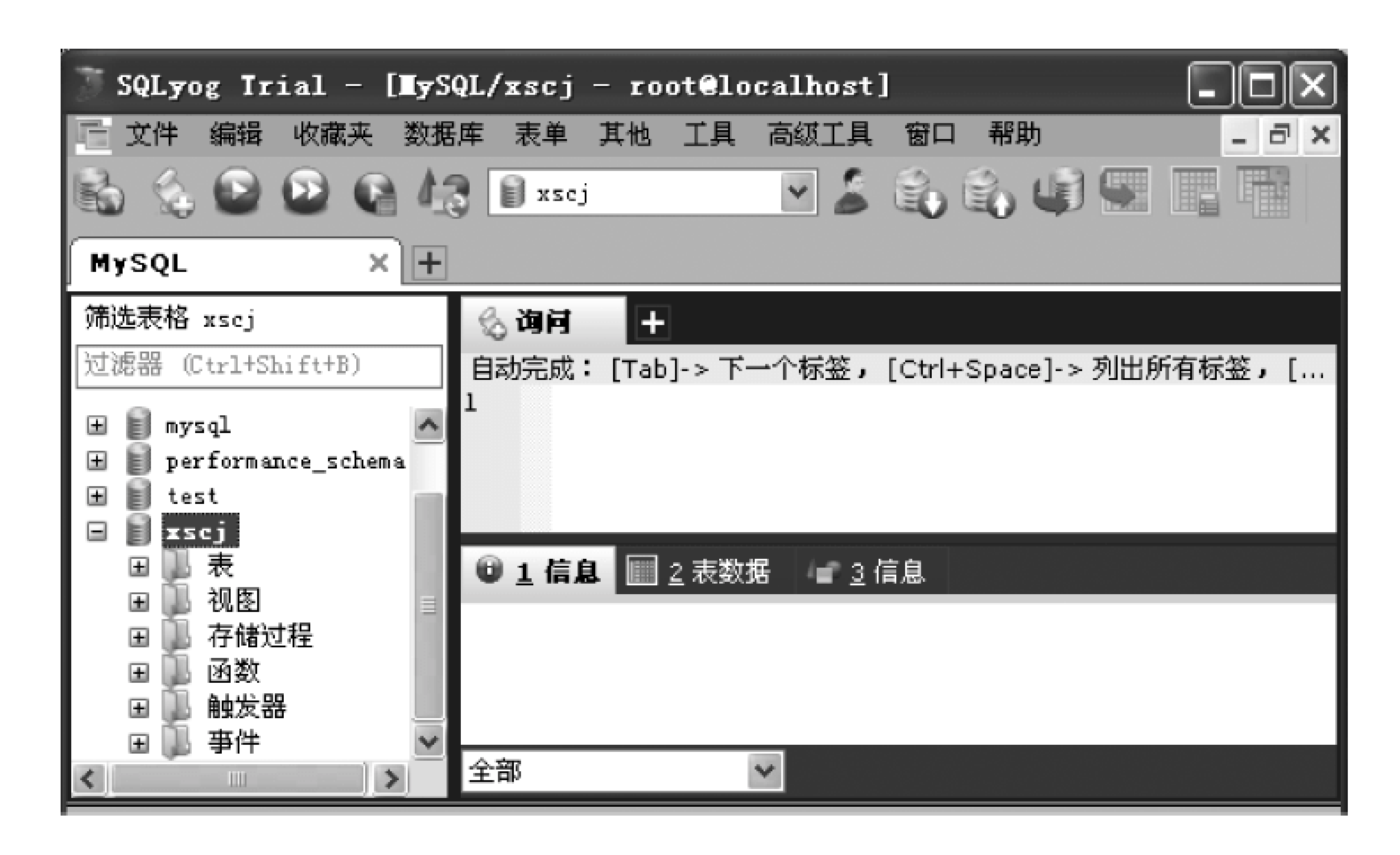


图 8.23　数据库创建成功

4. MySQL 界面工具表操作

1) 创建表

【例 8.12】　在 xscj 数据库中创建 xs 表,步骤如下。

(1) 在“对象资源管理器”窗口中,右击 xscj 数据库,然后选择“创/建”|“表”命令,如图 8.24 所示,则会打开“新表”窗口。

(2) 在打开的“新表”窗口中,设置相应信息,具体内容如图 8.25 所示。其中,“表名称”文本框中输入表的名称,“列”选项卡中的“列名”设置字段名,“数据类型”列设置字段的类型,“长度”列设置列类型的宽度。

(3) 最后在“新表”窗口中单击“保存”按钮,实现 xs 表的创建。在“对象资源管理器”窗口中选择 xscj 数据库,单击“刷新”按钮,则会在“表”结点中显示表 xs,如图 8.26 所示。

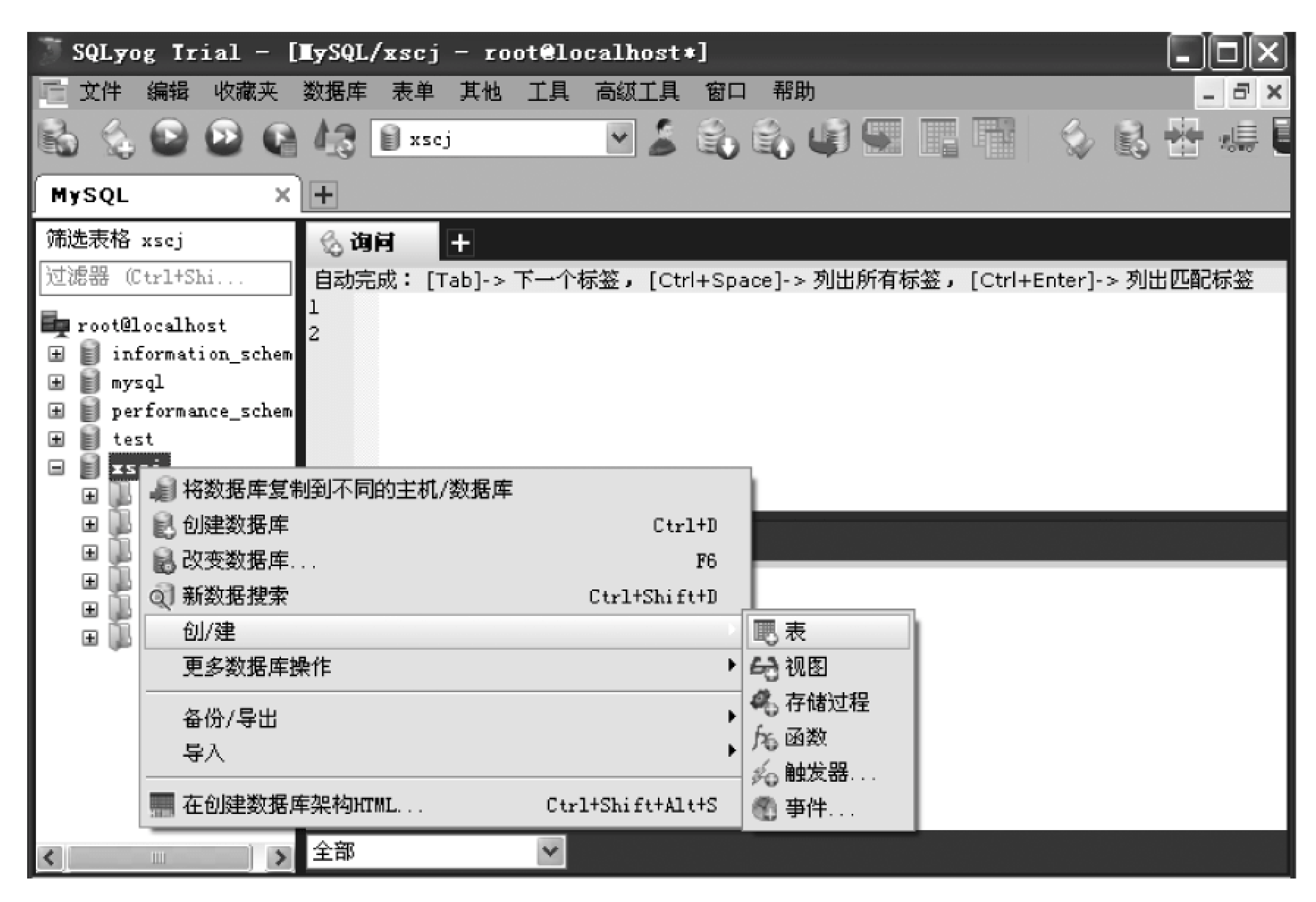


图 8.24　新建表

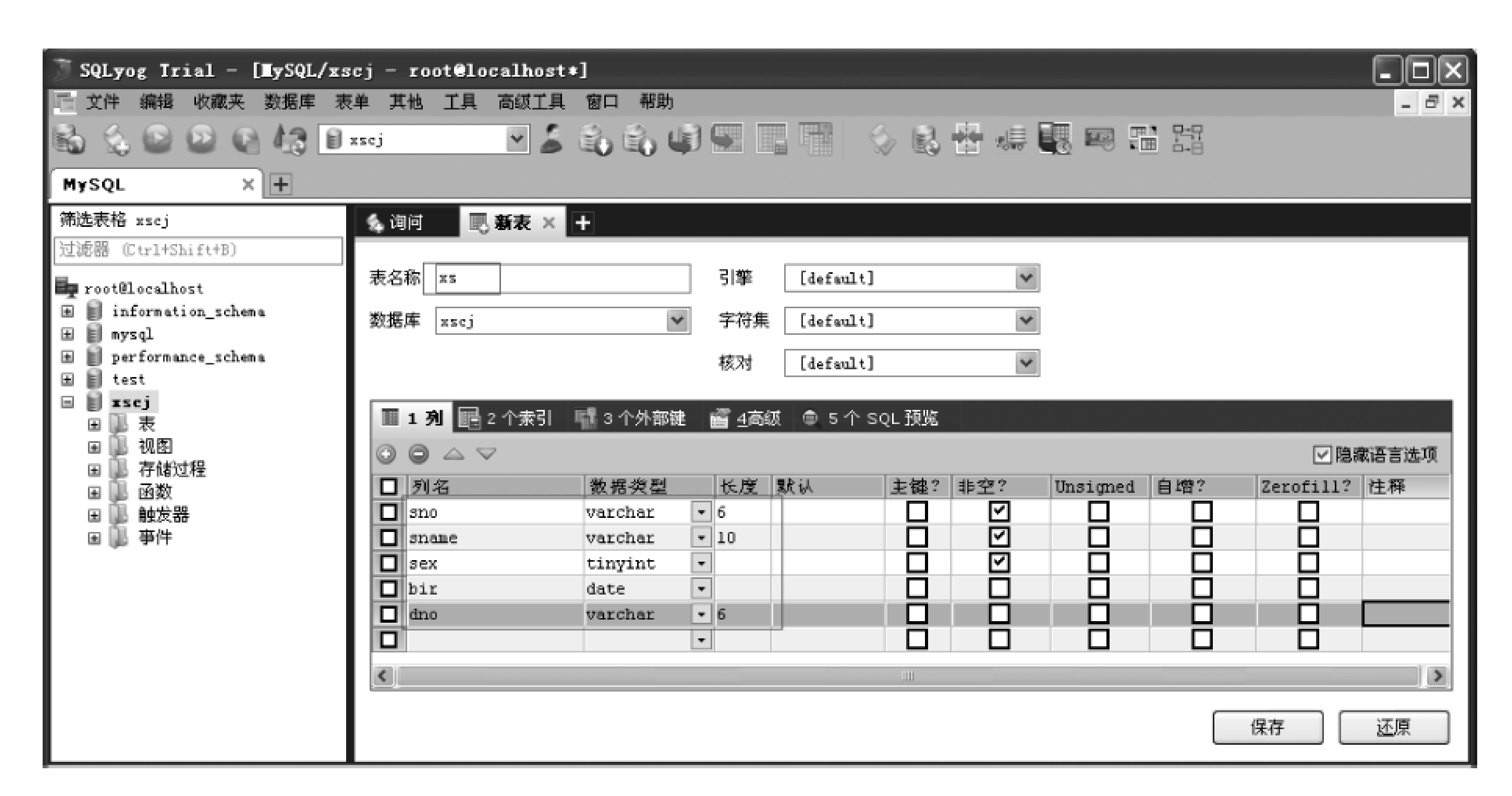


图 8.25　创建表的详细信息

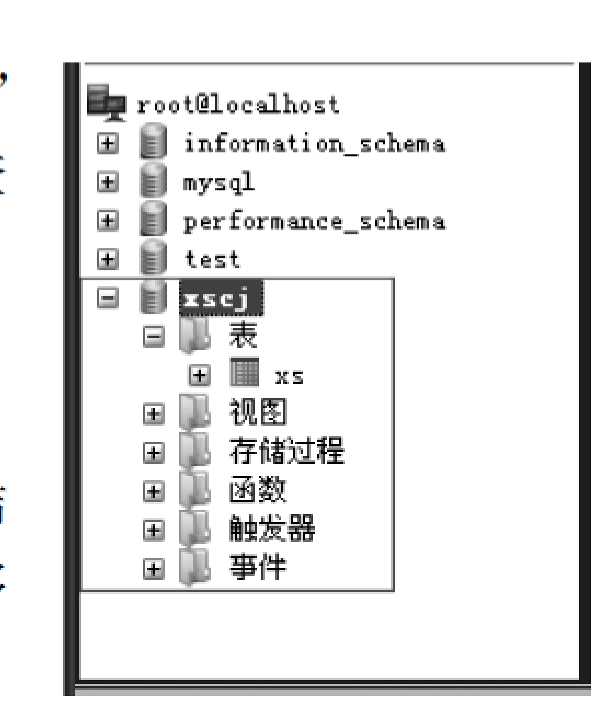


图 8.26　创建表成功

除了通过上述步骤(向导方式)创建表外,还可以在"询问"窗口中输入创建表的 SQL 语句,然后单击工具栏中的"执行查询"按钮 也可实现表的创建,如图 8.27 所示。

2) 删除表

如果要在数据库中删除表,步骤如下。

(1) 在"对象资源管理器"窗口中,单击数据库 xscj 中表结点前的加号,然后右击 xs 结点,从弹出的快捷菜单中选择"更多表操作"|"从数据库删除表"命令,如图 8.28 所示。

(2) 当单击"从数据库删除表"命令时,会弹出对话框确认

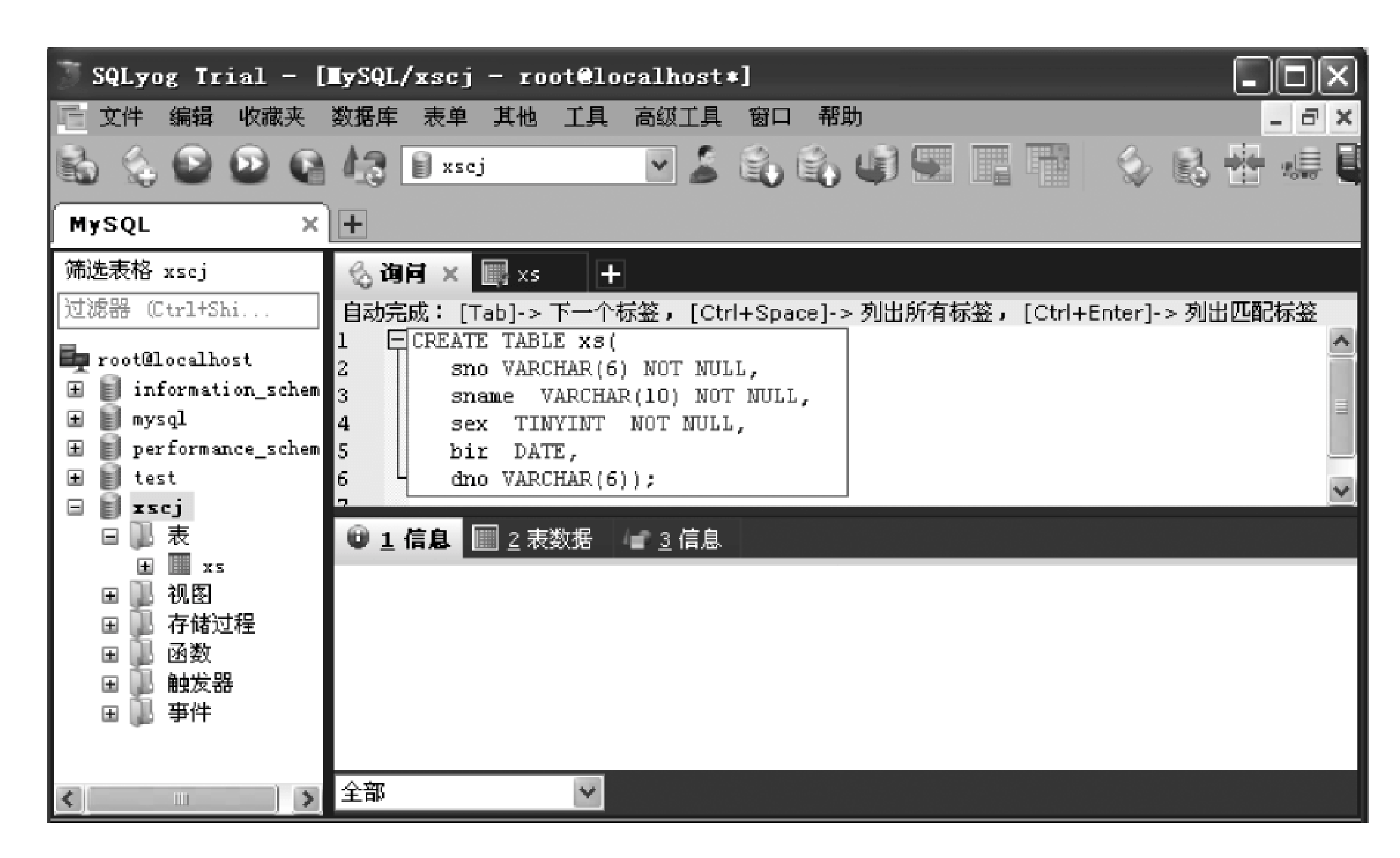


图 8.27　输入 SQL 语句创建表

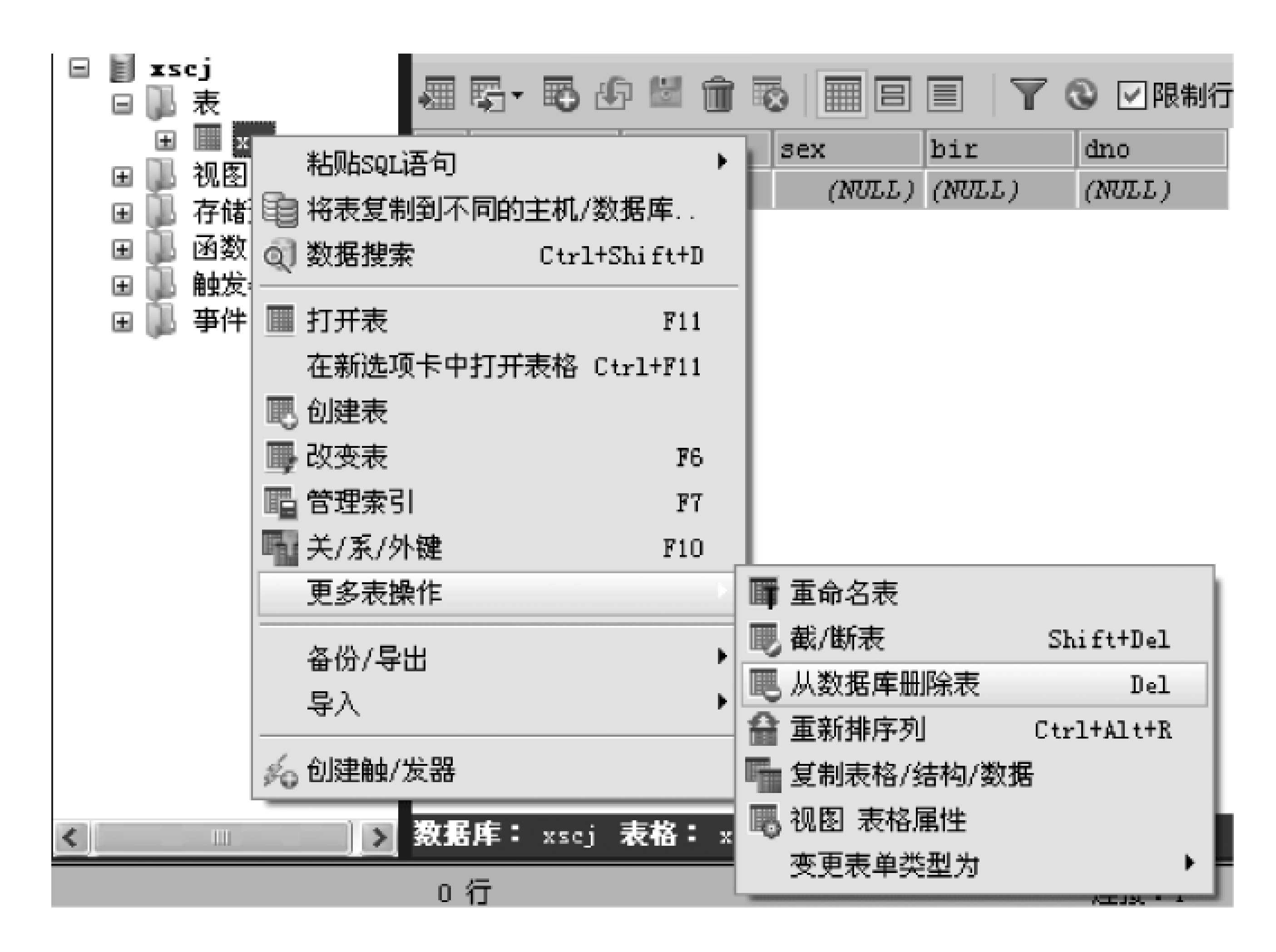


图 8.28　删除表命令

是否删除表，单击“是”按钮后，这时“对象资源管理器”窗口数据库 xscj 中“表”结点里就没有任何表对象。

如果要修改表，例如增加表中字段，可以在步骤(1)中选择“改变表”选项，在弹出的窗口中进行修改，修改后单击“保存”按钮即可完成。

3) 表中记录操作

添加记录的方法如下。

(1) 首先连接 MySQL 软件，在对象资源管理器窗口中选择 xscj 数据库，然后单击表结点，这时会显示数据库中的表格。

(2) 在对象资源管理器窗口中，右击 xs 表，然后选择“打开表”命令，如图 8.29 所示，则

会打开“表数据”窗口。



图 8.29 “表数据”窗口

(3) 在打开的“表数据”窗口中，可以直接输入数值，然后单击“保存”按钮，就可以实现数据记录值的录入，如图 8.30 所示。



图 8.30 插入数据记录

(4) 为了校验是否插入成功，可以单击表数据窗口中的“刷新”按钮，如果数据记录插入成功，则会在“表数据”窗口中显示出数据记录。

修改记录的方法和添加数据基本相同。在打开的“表数据”窗口中，会显示该表所包含的所有数据记录，如图 8.31 所示，双击需要修改的地方，输入新记录，修改完后单击“保存”按钮 ▣ 即可。

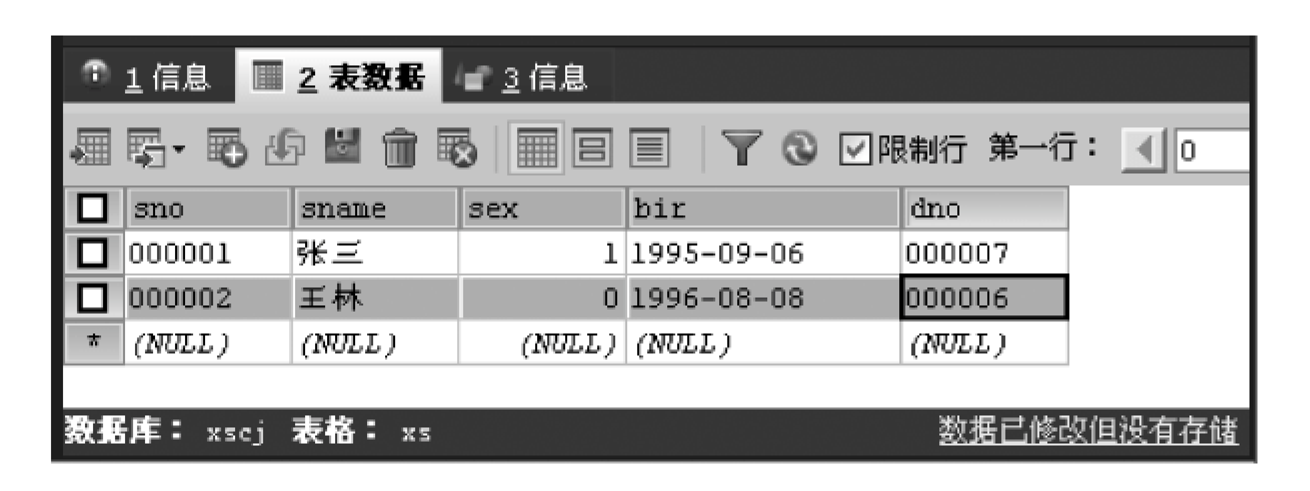


图 8.31　处于编辑状态

删除记录的方法是：在打开的“表数据”窗口中，会显示该表所包含的所有数据记录，然后找到所要删除的数据记录，找到后单击关于该数据记录的复选框，使该数据记录处于被选中状态，如图 8.32 所示。单击“删除”按钮 即可。这时会出现删除确认对话框，单击“是”按钮即实现选中数据记录的删除。删除后的结果如图 8.33 所示。

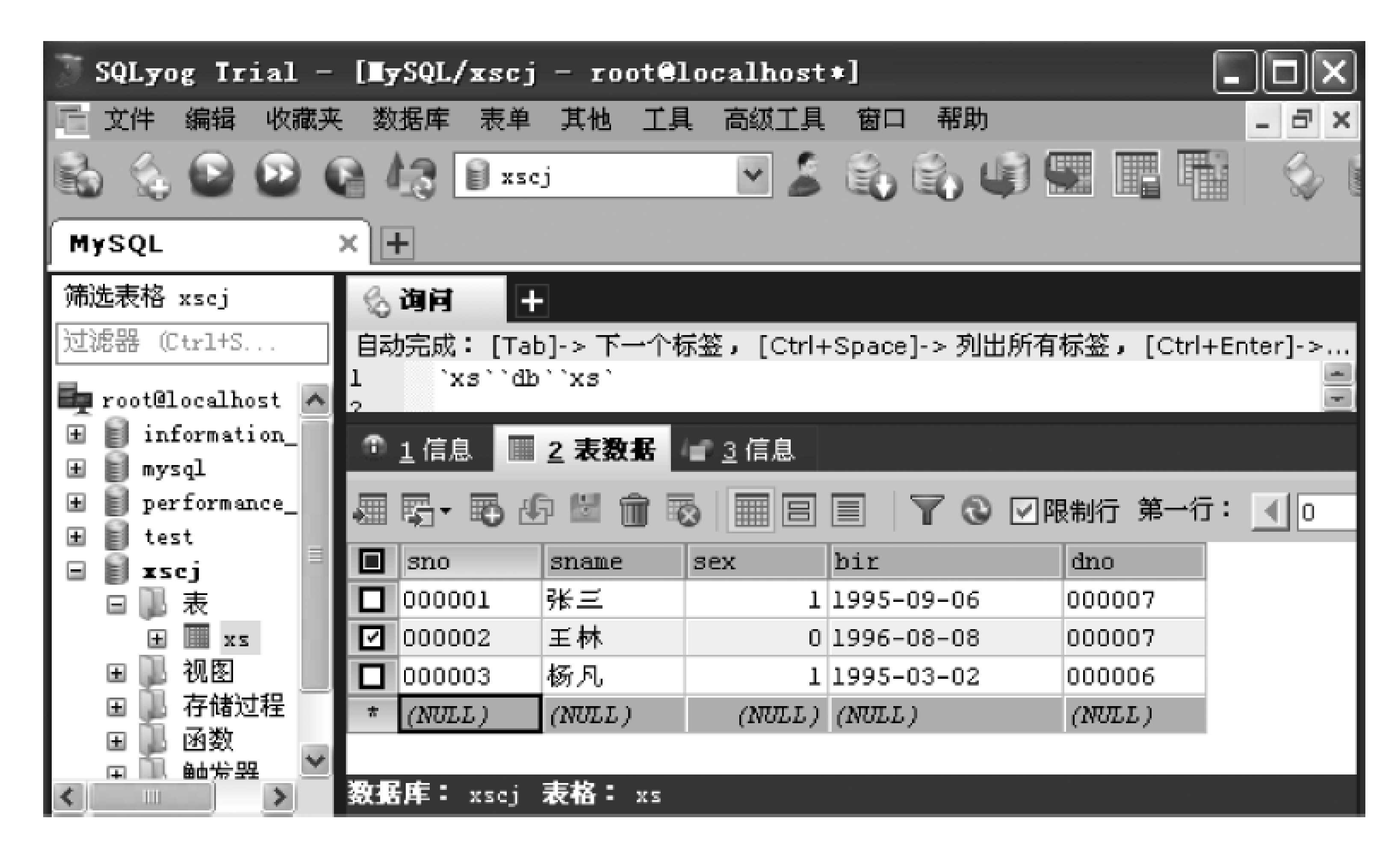


图 8.32　删除“选中”记录



图 8.33　删除记录成功

8.2.2 查询语句

对数据的基本操作包括数据插入、数据更新、数据删除、数据查询操作,前面介绍了插入数据记录操作、更新数据记录操作和删除记录操作,本节主要介绍查询数据记录操作。

查询数据记录,就是指从数据库对象表中获取所要求的数据记录,该操作是应用频率最高、最重要的操作。在 MySQL 数据库中,利用 SELECT 语句可以完成不同类型的复杂查询任务。SELECT 语句的语法格式如下:

```
SELECT [ALL | DISTINCT ] column_name[,expression… ]
[FROM table1_name [ , table2_name,view_name,] … ]          /* FROM 子句 */
[WHERE condition]                                           /* WHERE 子句 */
[ORDER BY {col_name | expr }[ASC | DESC] , … ]              /* ORDER BY 子句 */
[GROUP BY {col_name | expr } [ASC | DESC], … ]              /* GROUP BY 子句 */
[HAVING group_condition]                                    /* HAVING 子句 */
```

当执行一条 SELECT 语句时,系统会根据 WHERE 子句的条件表达式 condition,从 FROM 子句指定的基本表或视图中找出满足指定条件的记录,再按 SELECT 子句中的目标列或目标表达式,从数据库中选出匹配的行和列,并将结果存放到临时的表中。如果有 GROUP 子句,则将结果按特定列进行分组;如果 GROUP BY 子句带有 HAVING 子句,则只有满足指定条件 group_condition 的组才会返回;如果有 ORDER BY 子句,则对返回的结果进行排序。

为了便于讲解,本节所涉及的查询数据记录操作,都是针对 xscj 数据库中表示学生基本信息的表 xs,课程基本信息表 kc,学生选课信息表 sc,系基本信息表 dept,表的结构分别如图 8.34～图 8.37 所示。

```
mysql> select * from xs;
+--------+--------+-----+------------+--------+
| sno    | sname  | sex | bir        | dno    |
+--------+--------+-----+------------+--------+
| 130601 | 赵林林 | 女  | 1995-02-10 | 070001 |
| 130602 | 李芳   | 女  | 1994-10-03 | 070002 |
| 130603 | 张强   | 男  | 1995-08-08 | 070001 |
| 130604 | 王彦军 | 男  | 1994-06-10 | 070001 |
| 130605 | 罗琳   | 女  | 1994-11-01 | 070001 |
+--------+--------+-----+------------+--------+
5 rows in set (0.00 sec)

mysql>
```

图 8.34　xs 表数据

```
mysql> select * from kc;
+-----+------------------+------+--------+--------+
| cno | cname            | term | period | credit |
+-----+------------------+------+--------+--------+
| 101 | 计算机文化基础   |    1 |     64 |      4 |
| 102 | 面向对象程序设计 |    2 |     64 |      4 |
| 103 | 数据结构         |    3 |     64 |      4 |
| 104 | 操作系统         |    3 |     48 |      3 |
| 105 | 数据库原理       |    4 |     64 |      4 |
| 106 | 离散数学         |    3 |     80 |      5 |
+-----+------------------+------+--------+--------+
6 rows in set (0.00 sec)

mysql>
```

图 8.35　kc 表数据

```
mysql> select * from sc;
+--------+-----+-------+
| sno    | cno | grade |
+--------+-----+-------+
| 130601 | 101 |    85 |
| 130601 | 102 |    90 |
| 130602 | 101 |    90 |
| 130602 | 102 |    70 |
| 130603 | 101 |    87 |
| 130603 | 102 |    50 |
| 130605 | 101 |  NULL |
+--------+-----+-------+
7 rows in set (0.00 sec)

mysql>
```

图 8.36　sc 表数据

```
mysql> select * from dept;
+--------+------------------+----------------+
| dno    | dname            | dloc           |
+--------+------------------+----------------+
| 070001 | 计算机科学与技术系 | 11号楼12层1201 |
| 070002 | 网络工程系        | 11号楼12层1202 |
| 070003 | 信息技术系        | 11号楼12层1203 |
+--------+------------------+----------------+
3 rows in set (0.00 sec)

mysql>
```

图 8.37　dept 表数据

在 MySQL 中进行查询数据记录操作，具体步骤如下。

（1）执行 SQL 语句 USE，选择数据库 xscj，具体 SQL 语句如下：

```
USE xscj;
```

（2）执行 SQL 语句 SELECT，查询所有字段数据，具体 SQL 语句如下：

```
SELECT sno,sname,sex,bir,dno FROM xs;
```

执行上面的 SQL 语句，其结果如图 8.38 所示。

```
mysql> use xscj;
Database changed
mysql> SELECT sno,sname,sex,bir,dno FROM xs;
+--------+-------+-----+------------+--------+
| sno    | sname | sex | bir        | dno    |
+--------+-------+-----+------------+--------+
| 130601 | 赵林  |   1 | 1995-02-10 | 070001 |
+--------+-------+-----+------------+--------+
1 row in set (0.02 sec)

mysql> _
```

图 8.38　查询表数据

以下介绍各种查询语句的语法。

1．基本查询

基本查询主要指对单个表或视图进行无条件查询、有条件查询等。

1）无条件查询

在 SELECT 语句中，SELECT 子句后的目标列可以是表中的所有列、部分列，也可以是

表达式,包括算术表达式、字符串常量、函数等。

(1) 查询所有的列。如果查询表或视图中所有列的数据,可以用“ * ”表示目标列。

【例 8.13】 查询 xs 表中全体学生的基本信息。

```
SELECT *  FROM xs;
```

(2) 查询指定列。如果查询特定列的数据,可以在目标列中列出相应的列名,各列名之间要以逗号分隔。

【例 8.14】 查询 xscj 数据库的 kc 表中所有课程的课程编号、课程名。

```
SELECT cno,cname  FROM kc;
```

(3) 改变列标题。可以为查询的目标列或目标表达式起别名,即改变列标题。

【例 8.15】 查询 xs 表中学生的学号、姓名,结果中各列的标题分别指定为学号和姓名。

```
SELECT sno 学号,sname 姓名 FROM xs;
```

(4) 使用字符串常量。如果需要在查询结果中加入字符,可以在目标列表达式中使用字符串常量。

【例 8.16】 查询 xscj 数据库的 xs 表中的学号、姓名。结果集输出格式为:

```
sno'Name  is: '  sname
```

其中,sno 为查询结果集中第 1 列列名称,Name is:为查询结果集第 2 列列名称,sname 为查询结果集第 3 列列名称。

```
SELECT sno, 'Name  is: ',sname  FROM xs;
```

(5) 使用函数。可以在目标列表达式中采用函数对查询结果进行运算。

【例 8.17】 查询 xscj 数据库的 xs 表中全体学生的学号、姓名、出生年份。

```
SELECT sno, sname,YEAR(bir)  FROM xs;
```

其中,YEAR(date)返回 date 的年份(范围为 1000~9999)。

(6) 使用表达式。可以在目标列表达式中采用表达式对查询结果进行运算。

【例 8.18】 查询 xscj 数据库的 xs 表中全体学生的学号、姓名、年龄。

```
SELECT sno, sname,2014-YEAR(bir)  FROM xs;
```

查询结果如图 8.39 所示。

```
mysql> SELECT sno, sname,2014-YEAR(bir)  FROM xs;
+--------+--------+----------------+
| sno    | sname  | 2014-YEAR(bir) |
+--------+--------+----------------+
| 130601 | 赵林林 |             19 |
| 130602 | 李芳   |             20 |
| 130603 | 张强   |             19 |
| 130604 | 王彦军 |             20 |
| 130605 | 罗琳   |             20 |
+--------+--------+----------------+
5 rows in set (0.02 sec)

mysql>
```

图 8.39 使用表达式查询

(7) 清除重复行。如果不希望在查询结果中出现重复记录了，可以使用 DISTINCT 短语。

【例 8.19】 在 sc 表中查询学生选课学生的学生编号，结果集中不出现重复学生编号。

```
SELECT DISTINCT sno   FROM   sc;
```

2) 有条件查询

在执行无条件查询时，由于没有指定限制条件，所以会检索表或视图中的所有记录。在实际操作中，对数据的检索通常是有限制的，即有条件查询。有条件查询指在 SELECT 语句中使用 WHERE 子句设置查询条件，只有满足查询条件的记录才会返回。在 WHERE 子句中常用的运算符如表 8.2 所示。

表 8.2　WHERE 子句中常用的运算符

查询条件	运　算　符
比较运算	=,>,<,>=,<=,<>,!=
确定范围	BETWEEN AND,NOT BETWEEN AND
确定集合	IN, NOT IN
字符匹配	LIKE,NOT LIKE
空值判断	IS NULL,IS NOT NULL
逻辑操作	NOT,AND,OR

(1) 比较运算。比较运算符用于比较两个表达式的值，MySQL 支持的比较运算符有=(等于)、<(小于)、<=(小于等于)、>(大于)、>=(大于等于)、<>(不等于)、!=(不等于)。比较运算的语法格式为：expression { = | < | <= | > | >= | <=> | <> | != } expression。

当两个表达式值均不为空值(NULL)时，比较结果返回逻辑值 TRUE(真)或 FALSE(假)。

【例 8.20】 查询系编号为“070001”的学生的所有信息。

```
SELECT *   FROM xs   WHERE dno = '070001';
```

【例 8.21】 查询考试成绩有不及格的学生的学号。

```
SELECT sno   FROM sc   WHERE grade < 60;
```

(2) 确定范围。当要查询的条件是某个值的范围时，可以使用 BETWEEN 关键字。BETWEEN 关键字指出查询范围。格式为：

```
expression [ NOT ] BETWEEN expression1 AND expression2
```

当不使用 NOT 时，若表达式 expression 的值在表达式 expression1 与 expression2 之间(包括这两个值)，则返回 TRUE，否则返回 FALSE；使用 NOT 时，返回值刚好相反。

【例 8.22】 查询 xscj 数据库 xs 表中不在 1995 年出生的学生情况。

```
SELECT * FROM xsWHERE bir NOT BETWEEN '1995 - 1 - 1' and '1995 - 12 - 31';
```

(3) 确定集合。使用 IN 关键字可以指定一个值表，值表中列出所有可能的值，当与值表

中的任一个匹配时,即返回 TRUE,否则返回 FALSE。使用 IN 关键字指定值表的格式为:

```
expression IN ( expression [, …n])
```

【例 8.23】 查询所在系为“'070001(计算机科学与技术系)”、“070002(网络工程系)”或“070003(信息技术系)”的学生的情况。

```
SELECT * FROM xs WHERE dno IN  ('070001', '070002', '070003');
```

xs 信息表中只有系编号,本例按所在系编号查询。如果按系名查询,则要使用后面所介绍的多表查询。

(4) 字符匹配。如果要进行模糊查询,可以在 WHERE 条件中使用 LIKE 或 NOT LIKE 运算符。为了实现模糊查询,常使用特殊符号“_”和“%”,其中“%”代表 0 个或多个字符。“_”代表单个字符。LIKE 运算符表达式的格式为:

```
match_expression [ NOT ] LIKE match_expression
```

【例 8.24】 查询 xscj 数据库 xs 表中姓“王”的学生学号、姓名及性别。

```
SELECT sno,sname,sex  FROM XS WHERE sname LIKE '王%';
```

【例 8.25】 查询 xscj 数据库 xs 表中姓“王”且全名为两个汉字的学生学号,姓名,性别。

```
SELECT sno,sname,sex  FROM XS WHERE sname LIKE '王_';
```

(5) 空值判断。当需要判定一个表达式的值是否为空值时,使用 IS NULL 关键字,格式为:

```
expression IS [ NOT ] NULL
```

当不使用 NOT 时,若表达式 expression 的值为空值,返回 TRUE,否则返回 FALSE。当使用 NOT 时,结果刚好相反。

【例 8.26】 某些学生选修课程后没有参加考试,所以有选课记录,但没有考试成绩。查询 xscj 数据库 sc 表中缺少成绩的学生的学号和相应的课程号。

```
SELECT sno,cno  FROM SC WHERE grade IS NULL;
```

(6) 逻辑操作。WHERE 子句的查询条件还可以将多个判定运算的结果通过逻辑运算符(AND、OR 和 NOT)再组成更为复杂的查询条件。AND 用于组合两个条件,两个条件都为 TRUE 时值才为 TRUE。OR 也用于组合两个条件,两个条件有一个条件为 TRUE 时值就为 TRUE。NOT 表示对判定的结果取反。

【例 8.27】 查询 xs 表中所在系为 070001(计算机科学与技术系)且为女同学的基本信息。

```
SELECT * FROM xs WHERE dno = '070001' and sex = '女';
```

2. 连接查询(多表查询)

在数据库中,相关数据可能存储在不同的表中,因此需要从两个或多个表中获取数据。

连接查询就是指从多个表或视图中查询信息。把不同表的数据组合到一个表中叫做表的连接。连接查询分为内连接和外连接。内连接是指根据指定的连接条件进行连接查询，只有满足连接条件的数据才会出现在结果集中。

（1）相等内连接。使用等号（“＝”）指定连接条件的连接查询。进行比较的不同表中列的名称可以不同，但类型必须匹配。如果连接的表中有相同名称的列，则需要在列名前加表名，以区分是哪个表中的列。

【例 8.28】 查询计算机科学与技术系的学生学号、姓名、系编号、系名称，语句为：

```
SELECT sno, sname, xs.dno, dname from xs, dept
WHERE xs.dno = dept.dno and dept.dname = '计算机科学与技术系';
```

实例效果如图 8.40 所示。



```
mysql> select sno,sname,xs.dno,dname
    -> from xs,dept
    -> where xs.dno=dept.dno and
    -> dept.dname='计算机科学与技术系'
    -> ;
+--------+--------+--------+--------------------+
| sno    | sname  | dno    | dname              |
+--------+--------+--------+--------------------+
| 130601 | 赵林林 | 070001 | 计算机科学与技术系 |
| 130603 | 张强   | 070001 | 计算机科学与技术系 |
| 130604 | 王彦军 | 070001 | 计算机科学与技术系 |
| 130605 | 罗琳   | 070001 | 计算机科学与技术系 |
+--------+--------+--------+--------------------+
4 rows in set (0.01 sec)

mysql>
```

图 8.40 相等内连接查询

如果要在不同表中查询数据，则必须在 FROM 子句中指定多个表。多个表名之间用逗号隔开。

例如，在 xscj 数据库中需要查找选修了离散数学课程的学生的姓名和成绩，就需要将 xs、kc 和 sc 三个表进行连接，才能查找到结果。

【例 8.29】 查找 xscj 数据库中有学生选修过的课程名和课程号，使用如下语句：

```
SELECT DISTINC Tkc.cname 课程名, sc.cno 课程号
FROM kc, sc
WHERE kc.cno = sc.cno;
```

（2）自身连接。如果是在同一个表或视图中进行的连接查询，则称为自身连接。相当于同一个表作为两个或多个表使用。

【例 8.30】 查询与姓名“赵林林”在同一个系的学生的学号，姓名，语句为

```
SELECT xs1.sno, xs1.sname
FROM xs xs1, xs xs2
WHERE xs1.dno = xs2.dno and xs2.sname = '赵林林';
```

其中，xs1，xs2 为 xs 表的别名。

实例效果如图 8.41 所示。


```
mysql> SELECT xs1.sno, xs1.sname
    -> FROM xs xs1,xs xs2
    -> WHERE xs1.dno=xs2.dno and xs2.sname='赵林林';
+--------+--------+
| sno    | sname  |
+--------+--------+
| 130601 | 赵林林 |
| 130603 | 张强   |
| 130604 | 王彦军 |
| 130605 | 罗琳   |
+--------+--------+
4 rows in set (0.00 sec)

mysql>
```

图 8.41　自身连接查询

(3) 外连接。内连接只连接匹配的行,外连接是在内连接的基础上,将某个连接表中不符合连接条件的记录加入结果集中。即外连接查询会返回所操作表中至少一个表的所有数据记录。

根据结果集中所包含不符合连接条件的记录来源的不同,外连接分为左外连接(LEFT)、右外连接(RIGHT)和全外连接(FULL)。其中,左外连接是指在内连接的基础上,经连接操作符左侧表中不符合连接条件的记录加入结果集中,与之对应的连接操作符右侧表列用 NULL 填充。右外连接包含右边表的全部行(不管左边的表中是否存在与它们匹配的行),以及左边表中全部匹配的行。全外连接是包含左、右两个表的全部行,不管另外一边的表中是否存在与它们匹配的行。

外连接语句的格式如下:

```
SELECT column_name1[,column_name2 … ]
FROM join_tablename1   LEFT|RIGHT|FULL   [OUTER] JOIN join_tablename2
ON join_condition
```

上述语句中,column_name 表示所要查询的列名字,来源于所连接的表 join_tablename1 和 join_tablename2,关键字 OUTER JOIN 表示表进行外连接,参数 join_condition 表示进行连接的条件。

【例 8.31】 查询学生的选修情况,若某个学生没有选课,xs 表记录保存在结果集中,而在 sc 表的属性上填空值(NULL),这时就需要外连接。具体语句为:

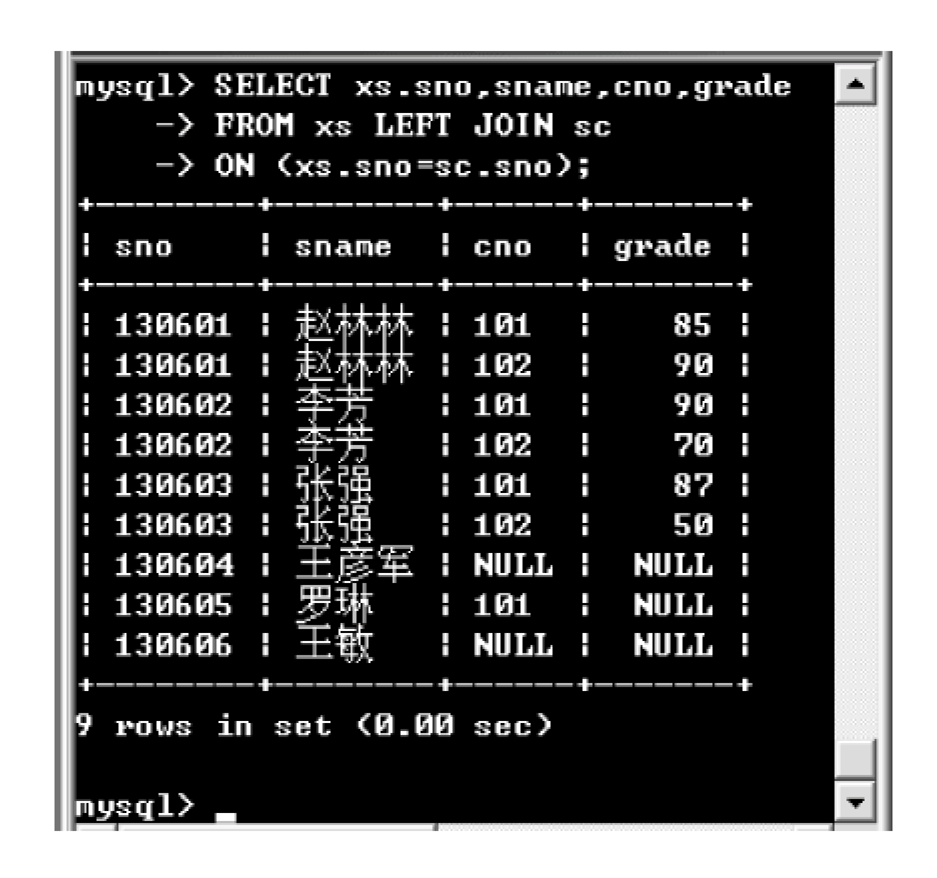
```
SELECT xs.sno,sname,cno,grade
FROM xs LEFT JOIN sc
ON (xs.sno = sc.sno);
```

运行效果如图 8.42 所示。

3. 查询排序

在执行查询操作时,可以使用 ORDER BY 子句对查询的结果进行排序。可以按升序或降序进行排序,可以针对一列或多列进行排序,也可以按表达式、别名等进行排序。

(1) 升序、降序排序。当使用 ORDER BY 子句对查询结果排序时,可以使用 ASC 或 DESC 设置按升序或降序排序,默认为升序排序。例如:


```
mysql> SELECT xs.sno,sname,cno,grade
    -> FROM xs LEFT JOIN sc
    -> ON (xs.sno=sc.sno);
+--------+--------+------+-------+
| sno    | sname  | cno  | grade |
+--------+--------+------+-------+
| 130601 | 赵林林 | 101  |    85 |
| 130601 | 赵林林 | 102  |    90 |
| 130602 | 李芳   | 101  |    90 |
| 130602 | 李芳   | 102  |    70 |
| 130603 | 张强   | 101  |    87 |
| 130603 | 张强   | 102  |    50 |
| 130604 | 王彦军 | NULL |  NULL |
| 130605 | 罗琳   | 101  |  NULL |
| 130606 | 王敏   | NULL |  NULL |
+--------+--------+------+-------+
9 rows in set (0.00 sec)

mysql> _
```

图 8.42　左外连接查询

```
SELECT sno,sname FROM xs ORDER BY sno;
SELECT sno,sname FROM xs ORDER BY sno DESC;
```

(2) 多列排序进行排序。对查询结果进行排序,也可以基于多列或多个表达式进行。当按多列或多个表达式进行排序时,首先按照第一列或表达式进行排序;当第一列或表达式的数据相同时,按第二列或表达式进行排序,以此类推。

【例 8.32】　查询学生信息,按学生所在系编号降序、学号升序排序,语句为:

```
SELECT sno,sname ,dno FROM xs ORDER BY dno DESC, sno;
```

效果如图 8.43 所示。


```
mysql> SELECT sno,sname ,dno FROM xs ORDER BY dno DESC, sno;
+--------+--------+--------+
| sno    | sname  | dno    |
+--------+--------+--------+
| 130606 | 王敏   | 070003 |
| 130602 | 李芳   | 070002 |
| 130601 | 赵林林 | 070001 |
| 130603 | 张强   | 070001 |
| 130604 | 王彦军 | 070001 |
| 130605 | 罗琳   | 070001 |
+--------+--------+--------+
6 rows in set (0.00 sec)

mysql> _
```

图 8.43　多列排序查询

(3) 使用列位置编号排序。当执行排序操作时,也可按照目标列或表达式的位置编号排序。如果列名或表达式名称很长,那么使用位置排序可以缩短排序语句的长度。

【例 8.33】　查询学生学号,姓名,年龄信息,查询结果按学号升序排序,按年龄降序排序。

```
SELECT sno,sname,2014-YEAR(bir) age  FROM xs ORDER BY sno,3 DESC;
```

4. 统计查询

(1) COUNT 函数。最经常使用的聚合函数是 COUNT()函数,用于统计组中满足条

件的行数或总行数,返回 SELECT 语句检索到的行中非 NULL 值的数目,若找不到匹配的行,则返回 0。语法格式为:

```
COUNT ( { [ ALL | DISTINCT ] expression } | * )
```

ALL 表示对所有值进行运算,DISTINCT 表示除去重复值,默认为 ALL。使用 COUNT(*)时将返回检索行的总数目,不论其是否包含 NULL 值。

【例 8.34】 求学生的总人数。

```
SELECT COUNT( * ) AS '学生总数'FROM xs;
```

(2) MAX 和 MIN 函数。MAX( )和 MIN( )函数分别用于求表达式中所有值项的最大值与最小值,语法格式为:

```
MAX / MIN ( [ ALL | DISTINCT ] expression )
```

其中,expression 是常量、列、函数或表达式。

【例 8.35】 求选修课程号为 101 的课程的学生的最高分和最低分。

```
SELECT MAX(grade), MIN(grade)FROM sc WHERE cno = '101';
```

(3) SUM 函数和 AVG 函数。SUM( )和 AVG( )函数分别用于求表达式中所有值项的总和与平均值,语法格式为:

```
SUM / AVG ( [ ALL | DISTINCT ] expression )
```

【例 8.36】 求学号为“130602”的学生所学课程的总成绩。

```
SELECT sno,SUM(grade) AS '课程总成绩'  FROM sc WHERE sno = '130602';
```

5. 分组查询

在数据查询过程中,经常需要将数据进行分组,以便对各个组进行统计分析。在 MySQL 数据库中,分组统计是由 GROUP BY 子句、集合函数、HAVING 子句共同实现的。

分组查询的基本语法:

```
SELECT column,group_function, …
FROM table
[WHERE condition]
[GROUP BY group_by_expression]
[HAVING group_condition]
[ORDER BY column[ASC|DESC]];
```

注意:

GROUP BY 子句用于指定分组列或分组表达式。

集合函数用于分组统计。如果未对查询分组,则集合函数将作用于整个查询结果;如果对查询结果分组,则集合函数将作用于每一个组,即每一个组都有一个集合函数。

HAVING 子句用于限制分组的返回结果。

WHERE 子句对表中的数据进行过滤,HAVING 子句对分组后形成的组进行过滤。

在分组查询中，SELECT 子句后面的所有目标列或目标表达式可以分组列或分组表达式或集合函数。

(1) 单列分组查询。将查询出来的记录按照某一个指定的列进行分组，分组列的值相等的记录为一组，然后对每一个组进行统计。

【例 8.37】 查询课程号及选课人数。

```
SELECT cno 课程编号, count(sno)选课人数 FROM sc
GROUP BY cno;
```

实例效果如图 8.44 所示。

```
mysql> SELECT cno 课程编号,count(sno) 选课人数 FROM sc
    -> GROUP BY cno;
+----------+----------+
| 课程编号 | 选课人数 |
+----------+----------+
| 101      |        4 |
| 102      |        3 |
+----------+----------+
2 rows in set (0.00 sec)

mysql>
```

图 8.44　单列分组查询

(2) 使用 HAVING 子句限制返回组。如果需要对分组后的查询结果做进一步限制，可以使用 HAVING 子句，只有满足条件的组才会返回。

【例 8.38】 查询选修了三门以上课程的学生学号。

```
SELECT sno FROM sc
GROUP BY sno
HAVING COUNT( * )> 3;
```

8.3　MySQL 数据库备份与恢复

尽管系统采取了各种措施来保证数据库的安全性和完整性，但硬件故障、软件错误、病毒、误操作或故意破坏仍可能发生，这些故障可能会破坏数据库，使数据库中的数据部分或全部丢失。因此数据库管理系统都提供了把数据库从错误状态恢复到某一正确状态的功能，这种功能称为恢复。恢复是以备份为基础的。数据库备份是最简单的保护数据的方法，MySQL 的备份和恢复组件为存在 MySQL 数据库中的数据提供了保护手段。

8.3.1　使用命令行备份和恢复数据库

在 MySQL 软件中经常通过命令实现数据备份，使用备份命令会将包含数据的表结构和数据内容保存在相应的文本文件中。具体执行时，首先会检查所需要备份数据的表结构，在相应的文本文件中生成 CREATE 语句。然后检查数据内容，在相应的文本文件中生成 INSERT INTO 语句。还原数据时，只需要执行文本文件中的 CREATE 语句和 INSERT INTO 语句。下面详细介绍命令行备份和恢复数据库。

1. 用 mysqldump 命令进行备份数据库

在 MySQL 软件中备份一个数据库通过命令 mysqldump 来实现,其命令形式如下:

```
mysqldump －h localhost －u username －p dbname table1 table2 … tablen > backupname.sql
```

上述语句中,-h 后面所跟的参数值指出备份的数据库在本机;-u 后面的 username 参数表示用户名;-p 指出密码;参数值 dbname 表示数据库;参数值 table 表示要备份的表,如果没有参数值 table,则表示备份整个数据库;参数值 backupname.sql 表示所生成的备份文件。

【例 8.39】 通过超级用户 root 登录到 MySQL 服务器,然后备份 xscj 数据库。

具体步骤:

(1) 打开 DOS 终端,进入 Bin 目录,路径为 C:\Program Files\MySQL\MySQL Server 5.5\bin。

(2) 执行命令 mysqldump,备份数据库 xscj,具体内容如下。

```
mysqldump －h localhost －u root －p123456 xscj > c:\xscj_2014.sql
```

运行效果如图 8.45 所示。

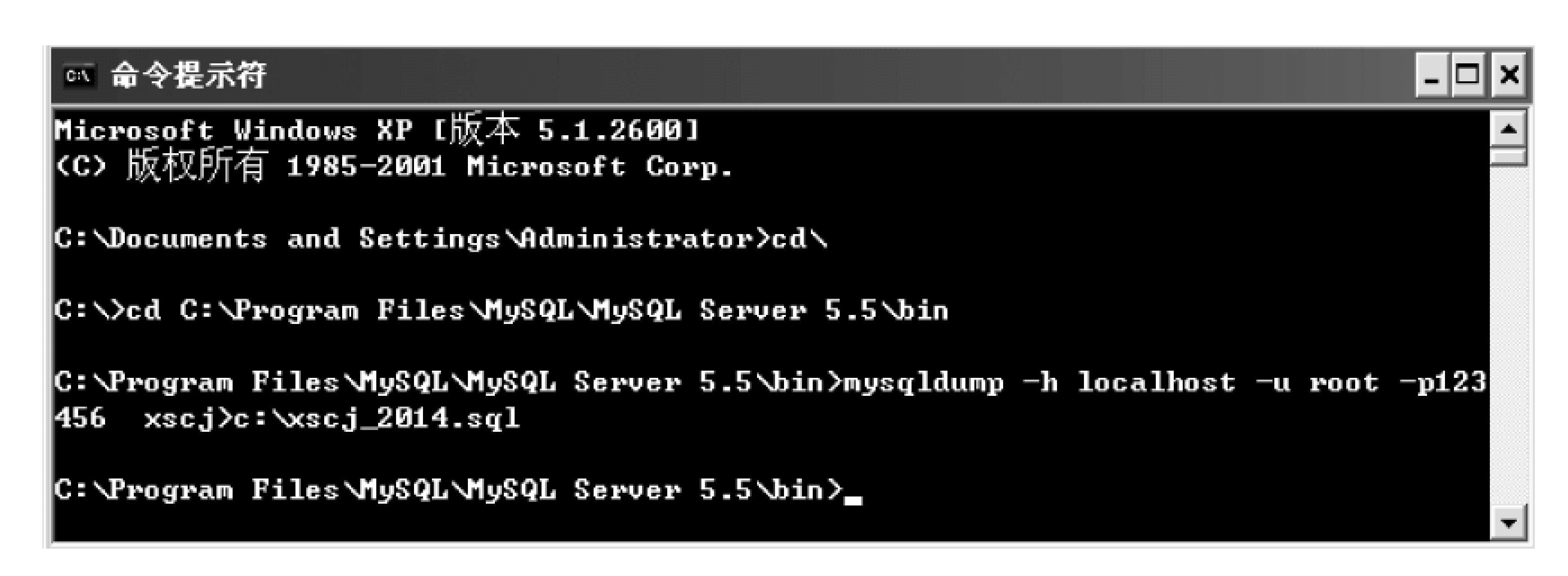


图 8.45 xscj 数据库的备份

说明:该命令结束没有分号;可以在命令里指定想要存放的目录,本例设定备份文件为 C 磁盘下的文件 xscj_2014.sql,结果生成的 xscj_2014.sql 文件会自动存在指定目录下;-h,-u 两个参数名与参数值之间可用或不用空格分隔,但-p 参数的参数名与参数值之间是一定不能加空格的,也可以这样输入:

```
mysqldump －hlocalhost －uroot －p xscj > xscj_2014.sql
```

执行该命令时,会显示"Enter password:",此时输入密码也可以实现成功备份。执行过程如图 8.46 所示。

```
命令提示符
C:\Program Files\MySQL\MySQL Server 5.5\bin>mysqldump -hlocalhost -uroot -p xscj
 >xscj_2014.sql
Enter password: ******

C:\Program Files\MySQL\MySQL Server 5.5\bin>
```

图 8.46 xscj 数据库的备份命令

(3) 打开磁盘 C,将创建一个名为 xscj_2014.sql 的新文件,具体内容如下。

```
-- MySQL dump 10.13 Distrib 5.5.37, for Win32 (x86)
--
-- Host: localhost    Database: xscj
-- ------------------------------------------------------
-- Server version5.5.37

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table 'dept'
--

DROP TABLE IF EXISTS 'dept';
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE 'dept' (
  'dno' varchar(6) NOT NULL,
  'dname' varchar(20) NOT NULL,
  'dloc' varchar(50) DEFAULT NULL,
  PRIMARY KEY ('dno')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table 'dept'
--

LOCK TABLES 'dept' WRITE;
/*!40000 ALTER TABLE 'dept' DISABLE KEYS */;
INSERT INTO 'dept' VALUES ('070001','计算机科学与技术系','11 号楼 12 层 1201'),('070002','网络工
程系','11 号楼 12 层 1202'),('070003','信息技术系','11 号楼 12 层 1203');
/*!40000 ALTER TABLE 'dept' ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table 'kc'
--

DROP TABLE IF EXISTS 'kc';
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE 'kc' (
  'cno' varchar(3) NOT NULL,
```

```
  'cname' varchar(20) NOT NULL,
  'term' int(11) DEFAULT NULL,
  'period' int(11) DEFAULT NULL,
  'credit' int(11) DEFAULT NULL,
  PRIMARY KEY ('cno')
) ENGINE = InnoDB DEFAULT CHARSET = utf8;
/ * !40101 SET character_set_client = @saved_cs_client * /;

--
-- Dumping data for table 'kc'
--

LOCK TABLES 'kc' WRITE;
/ * !40000 ALTER TABLE 'kc' DISABLE KEYS * /;
INSERT INTO 'kc' VALUES ('101','计算机文化基础',1,64,4),('102','面向对象程序设计',2,64,4),
('103','数据结构',3,64,4),('104','操作系统',3,48,3),('105','数据库原理',4,64,4),('106','离散
数学',3,80,5);
/ * !40000 ALTER TABLE 'kc' ENABLE KEYS * /;
UNLOCK TABLES;

--
-- Table structure for table 'sc'
--

DROP TABLE IF EXISTS 'sc';
/ * !40101 SET @saved_cs_client     = @@character_set_client * /;
/ * !40101 SET character_set_client = utf8 * /;
CREATE TABLE 'sc' (
  'sno' varchar(6) NOT NULL,
  'cno' varchar(3) NOT NULL,
  'grade' int(3) DEFAULT NULL,
  PRIMARY KEY ('sno','cno')
) ENGINE = InnoDB DEFAULT CHARSET = utf8;
/ * !40101 SET character_set_client = @saved_cs_client * /;

--
-- Dumping data for table 'sc'
--

LOCK TABLES 'sc' WRITE;
/ * !40000 ALTER TABLE 'sc' DISABLE KEYS * /;
INSERT INTO 'sc' VALUES ('130601','101',85),('130601','102',90),('130602','101',90),('130602','102
',70),('130603','101',87),('130603','102',50),('130605','101',NULL);
/ * !40000 ALTER TABLE 'sc' ENABLE KEYS * /;
UNLOCK TABLES;

--
-- Table structure for table 'xs'
--

DROP TABLE IF EXISTS 'xs';
/ * !40101 SET @saved_cs_client     = @@character_set_client * /;
/ * !40101 SET character_set_client = utf8 * /;
CREATE TABLE 'xs' (
```

```
  'sno' varchar(6) NOT NULL,
  'sname' varchar(10) NOT NULL,
  'sex' char(2) DEFAULT NULL,
  'bir' date DEFAULT NULL,
  'dno' varchar(6) DEFAULT NULL,
  PRIMARY KEY ('sno')
) ENGINE = InnoDB DEFAULT CHARSET = utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table 'xs'
--

LOCK TABLES 'xs' WRITE;
/*!40000 ALTER TABLE 'xs' DISABLE KEYS */;
INSERT INTO 'xs' VALUES ('130601','赵林林','女','1995-02-10','070001'),('130602','李芳','女','1994
-10-03','070002'),('130603','张强','男','1995-08-08','070001'),('130604','王彦军','男','1994
-06-10','070001'),('130605','罗琳','女','1994-11-01','070001'),('130606','王敏','女','1995
-02-10','070003');
/*!40000 ALTER TABLE 'xs' ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE = @OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE = @OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS = @OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS = @OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT = @OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS = @OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION = @OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES = @OLD_SQL_NOTES */;

-- Dump completed on 2014-07-03 17:51:03
```

2. 通过 mysql 命令实现数据还原

在 MySQL 软件中还原数据库通过 mysql 命令实现，其语法形式为：

```
mysql -u username -p [dbname] < backupname.sql
```

上述语句中，username 参数值表示用户名；参数 backupname.sql 表示所用来还原的备份文件；参数 dbname 为用户指定数据库的名称，可以指定也可以不指定，指定数据库时表示还原该数据库下的表，不指定数据库时表示还原备份文件中的所有数据库。

【例 8.40】 通过超级用户 root 登录到 MySQL 服务器，然后利用之前所生成的备份文件还原 MySQL 数据库。假设 xscj 数据库损坏，用备份文件进行恢复，前提：先新建 xscj 数据库。

具体步骤如下。

(1) 执行 SQL 语句 CREATE DATABASE，在数据库服务器里创建名为 xscj 的数据库。其结果如图 8.47 所示。

(2) 执行命令 mysql，还原数据库，具体命令如下：

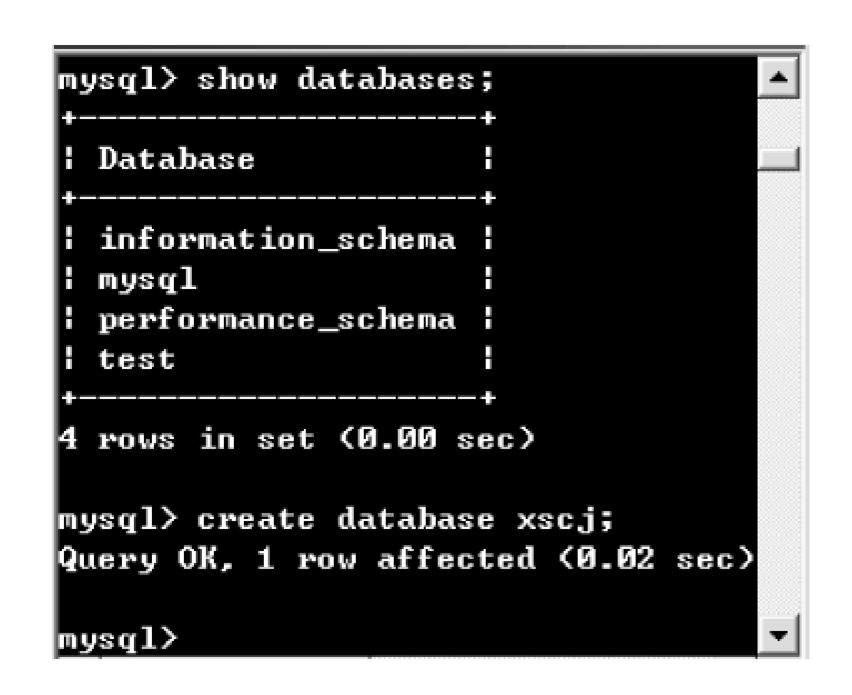


图 8.47 创建数据库

```
mysql  -hlocalhost  -uroot  -p123456  xscj< c:\xscj_2014.sql
```

输入此命令后,数据库中的数据将恢复到备份时的状态。其结果如图 8.48 所示。

```
命令提示符
C:\Program Files\MySQL\MySQL Server 5.5\bin>mysql  -hlocalhost  -uroot  -p123456
 xscj< c:\xscj_2014.sql

C:\Program Files\MySQL\MySQL Server 5.5\bin>_
```

图 8.48 还原数据库 xscj

(3) 为了校验是否成功,执行下面语句查看 xscj 数据库 xs 表中的内容,具体如图 8.49 所示。

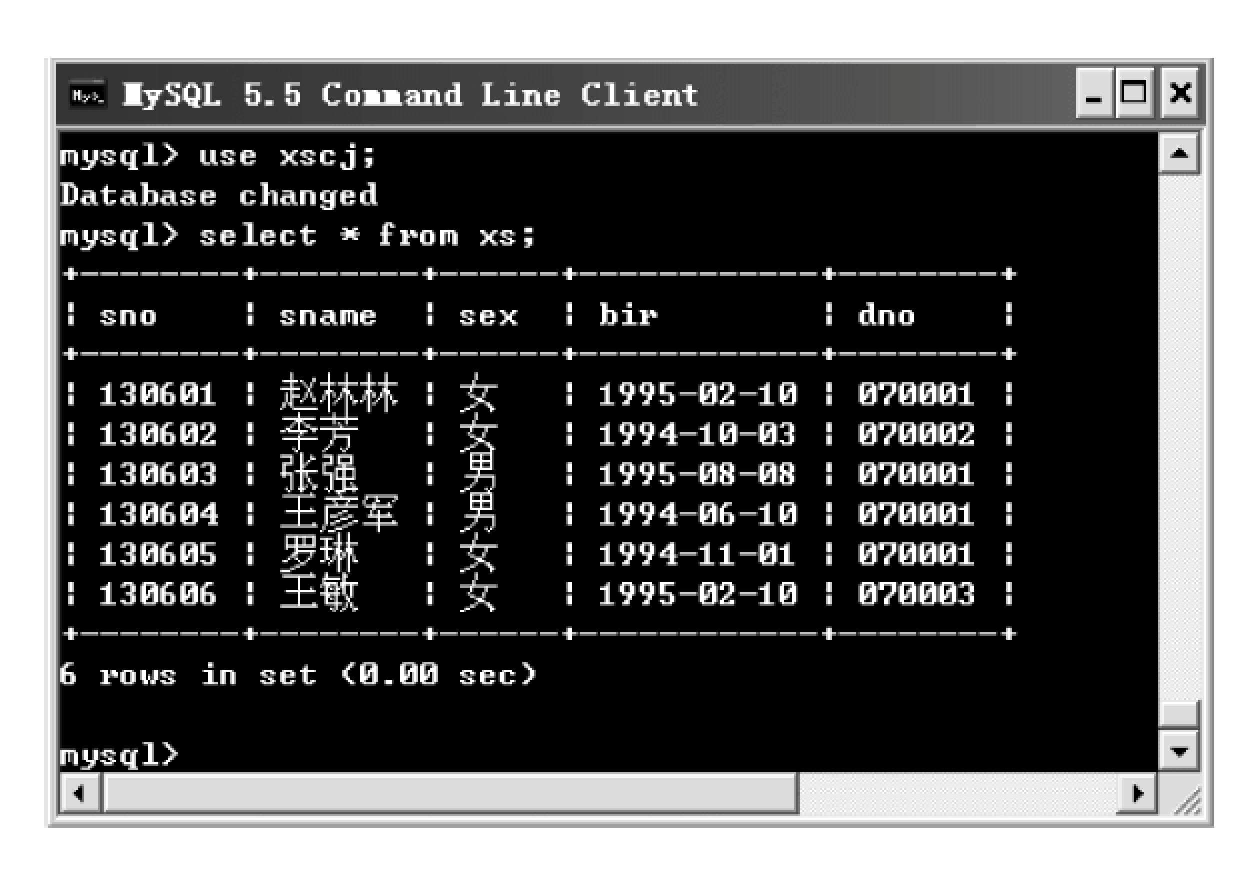


图 8.49 查询 xs 表中数据

注意:恢复时,xscj 数据库必须存在,数据库可空,恢复时会自动删除要恢复的表,再重新创建,插入数据。

8.3.2 使用 MySQL 界面工具进行备份和恢复

8.3.1 节通过 DOS 命令来实现数据库备份和恢复,本节通过具体实例来说明如何通过客户端软件 SQLyog 进行备份和恢复操作。

1. 通过客户端软件 SQLyog 进行备份数据库操作

【例 8.41】 同例 8.39,备份数据库 xscj。

具体步骤如下：

(1) 首先连接数据库服务器,在对象资源管理器窗口中显示 MySQL 数据库管理系统中所有的数据库,如图 8.50 所示。

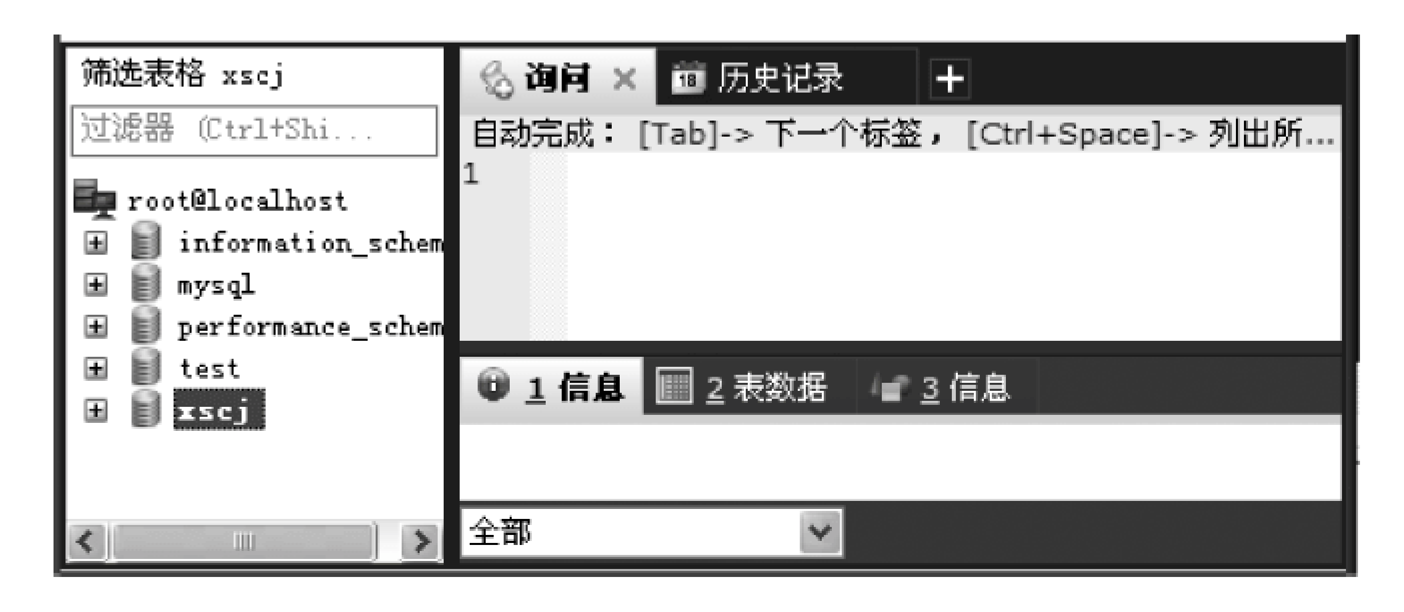


图 8.50 连接 MySQL 数据库管理系统

(2) 右击"对象资源管理器"窗口中数据库 xscj,在出现的菜单中选择"备份/导出"|"备份数据库,转储到 SQL…"命令,如图 8.51 所示,则会打开"以批处理脚本向导导出数据"对话框。

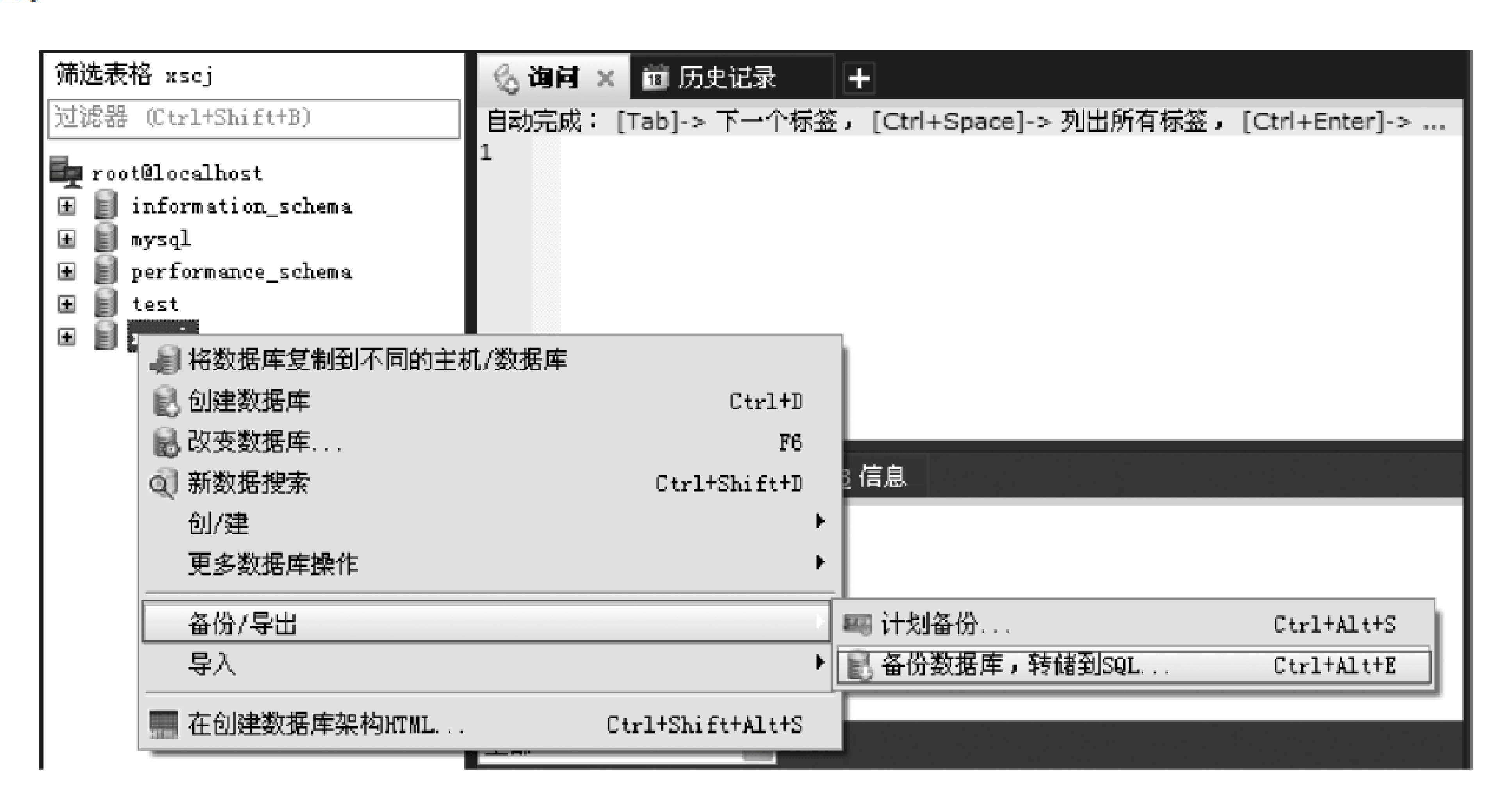


图 8.51 选择备份等命令

(3) 在"SQL 转储"对话框中,在"数据库名称"下拉列表框中选择相应的数据库,具体设置如图 8.52 所示,表示备份 xscj 数据库。单击"导出到文件"文本框右边 ... 按钮选择输入导出文件名称,若输入的文件不存在,则创建文件名为输入名称的文件。本例文件名称为 xscj_2014.sql。然后单击"导出"按钮。导出成功后,单击"完成"按钮,完成备份数据库操作,通过上述步骤,可以成功备份所选定的数据库。



图 8.52　备份 xscj 数据库

2. 通过客户端软件 SQLyog 进行还原操作

【例 8.42】　同例 8.40，通过备份文件还原数据库 xscj。

具体步骤如下：

(1) 首先连接数据库服务器，在“对象资源管理器”窗口中显示 MySQL 数据库管理系统中所有的数据库。单击 xscj 结点将显示该数据库下的所有表。

(2) 右击“对象资源管理器”窗口中数据库 xscj 结点，在出现的菜单中选择“导入”|“执行 SQL 脚本”命令，如图 8.53 所示，则会打开“从一个文件执行查询”对话框。



图 8.53　选择导入命令

(3) 在“从一个文件执行查询”对话框中,为“文件执行”对话框选择相应的备份文件,如图 8.54 所示,然后单击“执行”按钮就可以实现还原操作,当还原操作完成该对话框中的“关闭”按钮就会变成“完成”按钮,最后单击“完成”按钮则实现还原操作,如图 8.55 所示。

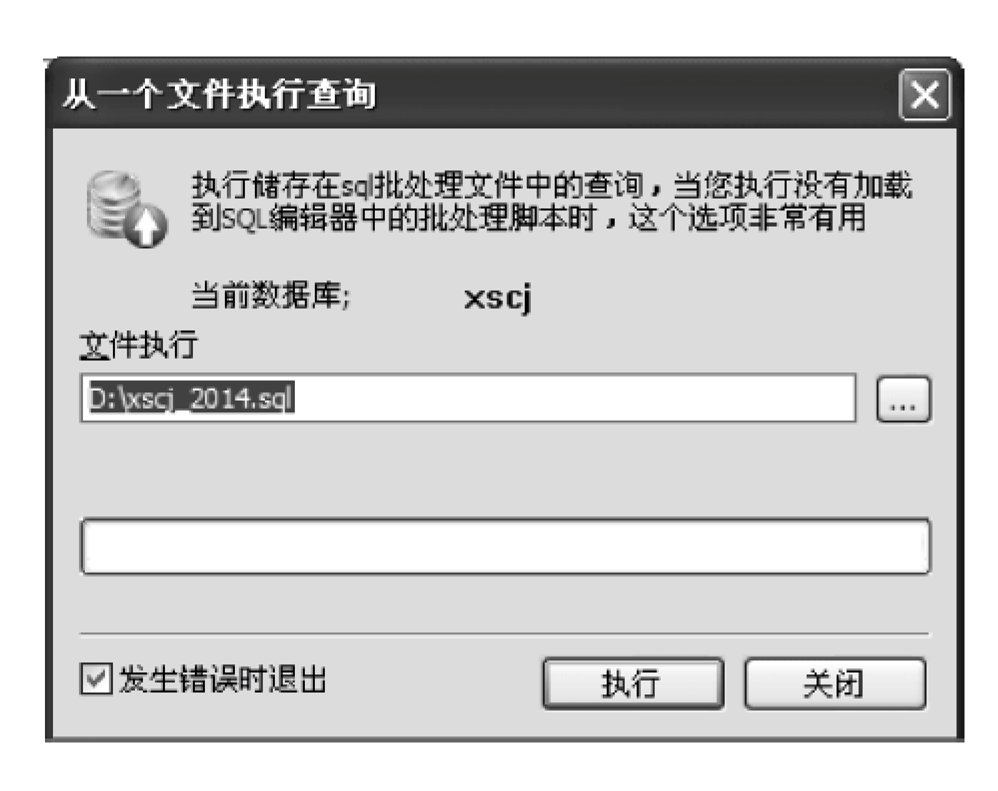


图 8.54　选择备份文件



图 8.55　还原操作完成

(4) 为了校验还原操作是否成功,单击“刷新对象浏览器”按钮,在“对象资源管理器”窗口中的 xscj 结点会显示数据库对象表。选择表 xs,然后单击“数据”选项卡标签,就会显示表中的数据。

通过上述步骤,可以从备份文件中还原数据库。

小结

本章详细介绍了目前常用的 MySQL 关系型数据库管理系统的使用,包括使用 SQL 在 MySQL 命令行客户端软件和常用的图形化管理软件(SQLyog 软件)中完成数据库及数据库对象——表的操作,表中数据的增删改查操作以及数据库的备份与恢复操作。重点需掌握 SQL 关系数据库语言、MySQL 关系型数据库管理系统的使用(包括数据库的基本操作、表的基本操作、数据查询和数据操作、数据库的备份和恢复),为进一步学习动态网站开发打好基础。



习题

1. 简述 SQL 的分类。

2. 按如表 8.3~表 8.5 所示结构利用 SQL 语句创建 xscj 数据库,并在该数据库中创建 xs、kc、sc 三个表。

表 8.3　xs 表

列名	数据类型	约束	备注
sno	varchar(6)	主键	学号
sname	varchar(10)		姓名
sex	varchar(2)		性别
bir	date		出生日期
dno	varchar(6)		所在系编号

表 8.4　kc 表

列名	数据类型	约束	备注
cno	varchar(6)	主键	课程编号
cname	varchar(10)		课程名称
term	int		开课学期
period	int		学时
credit	int		学分

表 8.5　sc 表

列名	数据类型	约束	备注
sno	varchar(6)	主键	学号
cno	varchar(6)	主键	课程编号
grade	int		成绩

3. 为 xs 表添加总学分和备注两列。总学分数据类型为整数类型;备注数据类型为文本类型。

4. 在 xscj 数据库的表 xs 中插入如下信息:学号:130601,姓名:赵林林,性别,出生日期:1995-02-10,所在院系编号:070001。

5. 将 xscj 数据库 xs 表中学号为“130601”学生的姓名改为“赵林”。

6. 使用 select 语句实现例 8.13~例 8.38 题目查询要求。

7. 使用命令行对 xscj 数据库进行备份。

8. 假设 xscj 数据库损坏,利用前面做过的备份文件使用命令行对 xscj 数据库进行恢复,并验证恢复是否成功。

9. 使用 MySQL 界面工具对 xscj 数据库进行备份。

10. 假设 xscj 数据库损坏,利用前面做过的备份文件使用 MySQL 界面工具对 xscj 数据库进行恢复,并验证恢复是否成功。

第9章 PHP语法基础

本章要点：

- PHP 基本语法；
- PHP 中的数据类型；
- 运算符与表达式；
- PHP 字符串；
- 使用数组；
- 流程控制；
- 使用函数使代码模块化。

PHP(Hypertext Preprocessor)语言是一种易于学习和使用的，用于创建动态交互性网站的服务器端脚本语言，由于其类似C语言的简洁语法、丰富的组件扩展以及完全免费等特性，自从20世纪90年代中期问世以来，就受到了Web开发人员的热烈欢迎，被广泛应用在各种类型的网站建设中。

9.1 PHP 基本语法

9.1.1 PHP 标记风格

在PHP程序中，通常会包含HTML标签。为了让PHP解释器能够在运行代码时将两者区分，可以采用以下几种格式来表示PHP代码：

```
<? … ?>
<?php … ?>
<script language="php"> … </script>
<% … %>
```

在编写PHP代码时，可以选用其中一种代码格式。不过需要注意的是，如果用第一种方式<?和?>来开始和结束脚本块这种写法，只能在支持简写的服务器上运行。因此，为了达到最好的兼容性，在正式的部署环境中建议使用标准形式(<? php)，而不用简写形式。

与Perl和C一样，PHP中的每个代码行都必须以分号(;)结束。分号是一种分隔符，用于把指令集区分开来。

9.1.2 PHP 程序注释

PHP 同时支持 C,C++和 UNIX 风格的注释方式:

```
/* C,C++风格多行注释 */
// C++风格单行注释
# UNIX 风格单行注释
```

一般来说,在 PHP 中,使用//来编写单行注释,使用/*和*/来编写一些大的注释块。

9.1.3 使用 PHP 输出 HTML

PHP 代码是服务器脚本,在后台运行,如果想要将运行数据在浏览器输出,必须使用自带的显示函数输出。一般使用 echo( )和 print( )这两种函数。

在例 9.1 中,运用上面所学的知识编写了一段简单的 PHP 脚本,它可以向浏览器输出文本“Hello World”。在此例中,使用了 echo 语句来输出文本“Hello World”。

【例 9.1】 在浏览器中输出文本“Hello World”。

```
<HTML>
 <HEAD>
 </HEAD>
 <BODY>
    <H1>
    First PHP page
    </H1>
    <HR>
    <?
    // Single line C++style comment
    /*
    printing the message
     */
    echo "Hello World!";
    # Unix style single line comment
    ?>
 </BODY>
 </HTML>
```

9.2 PHP 中的数据类型

PHP 的值的数据类型是变量的数据类型,在 PHP 中变量不需要声明就可以直接赋值,赋值的类型就是变量的类型。

PHP 支持 6 种基本的数据类型,包括整数 integer、浮点数 float、字符串 string、数组 array、对象 object 和布尔型 boolean。另外,PHP 还支持两种特殊的数据类型 NULL(空)和 resource(资源)。常见的数据类型如表 9.1 所示。

表 9.1　数据类型表

数据类型	简　　介	示　　例

数值	数值类型可以是整数或是浮点数	$ a = 1234;　　# 十进制数
		$ a = -123;　　# 负数
		$ a = 0123;　　# 八进制数(等于十进制数的 83)
		$ a = 0x12;　　# 十六进制数(等于十进制数的 18)
		$ a = 1.234;　　# 浮点数"双精度数"
		$ a = 1.2e3;　　# 双精度数的指数形式

字符串	字符串可以由单引号或双引号引出的字段定义。注意不同的是被单引号引出的字符串是以字面定义的，而双引号引出的字符串可以被扩展。反斜杠(\)可以被用来分隔某些特殊字符	$ first = 'Hello';
		$ second = "World";
		$ full1 = "$ first $ second"; # 产生 Hello World
		$ full2 = '$ first $ second'; # 产生 $ first $ second

数组	PHP 提供数组和哈希表的支持，可以用 list()或者 array()来定义它们，也可以直接为数组赋值。数组的索引从 0 开始	//一个包含两个元素的数组
		$ myarray = array (
		"key1" =>"00001",
		"key2" =>"00002",
		"key3" =>"00003"
		);

对象	使用 new 语句产生一个对象	class foo
		{
		function do_foo ()
		{
		echo "Doing foo.";
		}
		}
		//实例化该类型，并调用相应的方法
		$ bar = new foo;
		$ bar->do_foo();

在 PHP 中，变量会在使用时被自动声明。因此，PHP 通常被称作松散类型的语言(Loosely Typed Language)。

例如，变量 $ abc=123，在进行数学运算时，将被当作整数处理，而在进行 echo 输出时，则会自动被当成字符串处理。

例 9.2 是一段 PHP 中典型的声明变量的例子。

【例 9.2】 声明变量。

```
<?php
$ txt = "Hello World!";
$ number = 16;
?>
```

在例 9.2 中,创建了一个存有字符串的变量 $ txt 和一个存有数值的变量 $ number,可以看到在这个过程中,不需指定任何变量的数据类型。PHP 的这种特性大大简化了开发人员编写代码的难度,提高了程序的开发效率。

当然,根据需要,开发人员也可以通过函数 cast 或者 settype 来对数据类型进行明确的设定或强制类型转换。

PHP 中允许的强制类型转换有以下几种。

(int),(integer)——转换成整型

(bool),(boolean)——转换成 bool 型

(float),(double),(real)——转换成浮点型

(string)——转换成字符串

(array)——转换成数组

(object)——转换成对象

9.3 运算符与表达式

9.3.1 PHP 变量

在程序运行时,存储在内存中的数据有两种形态。在程序运行中不改变的值称为常量,根据条件发生变化的值称为变量。

PHP 中变量都是以美元符号“ $ ”标记的变量名来表示的,定义变量时要遵守一定的命名规则。

(1) 变量名必须以字母或下划线“_”开头。

(2) 变量名只能包含字母、数字字符以及下划线。

(3) 变量名不能包含空格。如果变量名由多个单词组成,那么应该使用下划线进行分隔(比如 $ my_string),或者以大写字母开头(比如 $ myString)。

通常将变量用于存储值,比如数字、字符串或函数的结果,这样就可以在 PHP 脚本的运算过程中多次使用它们了。

在 PHP 中,符号“=”不表示相等,而表示赋值,是将一个值指定给一个变量,它表示把“=”右边的表达式的值赋给左边的变量,如“ $ a=5”表示将 5 赋给 $ a。因此,在 PHP 中设置变量的正确方法是:

```
$ var_name = value;
```

9.3.2 PHP 运算符

PHP 具有 C 和 C++中的通常见到的运算符,这些运算符的优先级也是一致的。PHP 运算符包括:算术运算符、赋值运算符、比较运算符、逻辑运算符等。

PHP 常用的算术运算符如表 9.2 所示。

表 9.2　PHP 常用运算符

运算符	说明	例子	结果
+	加	x=2 x+2	4
-	减	x=2 5-x	3
*	乘	x=4 x*5	20
/	除	15/5 5/2	3 2.5
%	取模	5%2 10%8 10%2	1 2 0
++	递增	x=5 x++	x=6
--	递减	x=5 x--	x=4

【例 9.3】 变量赋值运算。

```
<?php
 $a=5;
 $b=3;
 $num= $a+ $b;
echo $num;                                    //输出 8
?>
```

PHP 常用的赋值运算符如表 9.3 所示。

表 9.3　PHP 赋值运算符

运算符	例子	结果
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=xy
%=	x%=y	x=x%y

【例 9.4】 字符串赋值。

```
<?php
 $a=10;
 $a+=3;
echo $a;                                      //输出 13
 $a*=2;
echo $a;                                      //输出 26
```

```
$string="你好";
$string.="中国";
echo $string;                                   //输出 你好中国
?>
```

PHP 常用的比较运算符如表 9.4 所示。

表 9.4 PHP 比较运算符

运算符	说明	例子
==	是否相等	5==8
!=	是否不相等	5!=8
>	是否大于	5>8
<	是否小于	5<8
>=	是否大于等于	5>=8
<=	是否小于等于	5<=8

PHP 常用的逻辑运算符如表 9.5 所示。

表 9.5 PHP 逻辑运算符

运算符	说明	例子
&&	与	x=6 y=3 (x<10 && y>1)
||	或	x=6 y=3 (x==5 || y==5)
!	取反	x=6 y=3 !(x==y)

9.4 PHP 字符串

9.4.1 字符串的定义与显示

无论何种程序语言,字符串操作都是一个重要的基础。PHP 提供了大量的字符串操作函数,在 PHP 中,一般用双引号或单引号来标识一个字符串。

【例 9.5】 字符串赋值。

```
<?php
$str1="你好";
echo 'str1 中国';                               //输出 str1 中国
echo "str1 中国";                               //输出 你好中国
?>
```

在 PHP 程序处理字符串中,单引号中的内容都会当作字符处理,而双引号中的特殊字

符则会被转义，因此上面的两个语句输出结果并不一致。简单地说，可以认为双引号里的内容是经过“解释”的，单引号里的则是“所见即所得”的。

PHP 里的输出最常用的是 echo()和 print()。这两者都不是真正的函数，而是语言构造，所以调用时不必用双括号。在输出的时候两者都可以实现赋值。两者除了名字不一样外，还是有其他区别的。print 具有返回值，一直返回 1，而 echo 没有，所以 echo 比 print 要快一些。也正因为这个原因，print 能应用于复合语句中，而 echo 不能。此外，echo 一次可输出多个字符串，而 print 则不可以。

定义好字符串后，就可以对字符串进行处理了。PHP 字符串操作函数，不仅功能强大，使用也比较简单。

9.4.2 常用字符串函数

1. 计算字符串长度

PHP 提供 strlen()函数来计算字符串的长度。需要注意，strlen()将中日等汉字以及全角字符都当作两个或四个长度计算。

语法格式如下：

```
Int strlen(string $ string)
```

【例 9.6】 计算字符串长度。

```
<?php
 $ str1 = "hello";
echo strlen(str1);                              //输出 6
 $ str2 = "你好";
echo strlen(str2);                              //输出 4
?>
```

2. 连接操作符

两个以上的字符串连接用“.”操作符，依字符串的顺序形成新的字符串。

【例 9.7】 连接字符串。

```
<?php
 $ str  =  "hello" . "world" ;
echo  $ str                                     //输出 helloworld
?>
```

3. 字符串的裁剪

一个字符串首和尾，可能有不想要的部分，可以用 trim()、rtrim()、ltrim()等函数，分别去除一个字符串两端空格，一个字符串尾部空格，一个字符串首部空格。

【例 9.8】 裁剪字符串。

```
<?php
echo trim(" hello,world");                      // 输出"hello,world"
```

```
echo rtrim(" hello,world ");                    // 输出"hello,world"
echo ltrim(" hello,world");                     // 将得到 "hello,world"
?>
```

4. 转换大小写函数

对于英文字母来说,可以用 strtoupper()、strtolower()函数将其转变成大写或小写。

【例 9.9】 转换字符串大小写。

```
<?php
echo strtoupper("i love you");                  // 输出 I LOVE YOU
echo strtolower("I LOVE YOU");                  // 输出 i love you
?>
```

5. 比较函数

PHP 中一般可以用!=,==直接比较两个字符串是否相等。此外,PHP 还提供了一组专门进行字符串比较的函数:strcmp(),strcasecmp(),strncasecmp(),strncmp()。比较后,如果前者比后者大,则都返回大于 0 的整数;如果前者比后者小,则都返回小于 0 的整数;如果两者相等,则返回 0。

strcmp()函数是用于区分大小写(即大小写敏感)的字符串比较。

strcasecmp()函数用于不区分大小写的字符串比较。

strncmp()函数用于比较字符串的一部分,从字符串的开头开始比较,第三个参数为要比较的长度。

strncasecmp 用于不区分大小写的比较字符串的一部分,从字符串的开头开始比较,第三个参数为要比较的长度。

【例 9.10】 比较字符串。

```
<?php
echo strcmp("abcdd", "aBcde");                  //输出 1 (>0), 比较的是 "b"和"B"
echo strcasecmp("abcdd", "aBcde");              //输出 -1 (<0), 比较的是"d"和"e"
echo strncmp("abcdd", "aBcde", 3);              //输出 1 (>0), 比较了 abc 和 aBc
echo strncasecmp("abcdd", "aBcde", 3);          //输出 0, 比较了 abc 和 aBc
?>
```

6. 替换函数

替换是指将一个字符串的一部分进行改变,使之成为另外一个新的字符串,以满足新的要求。PHP 里通常用 str_replace()函数进行字符串的替换。str_replace 是大小写敏感的。str_replace 还可以实现多对一,多对多的替换,但无法实现一对多的替换。

此外,PHP 还提供了 substr_replace(),实现替换一部分的字符串。

语法格式如下:

```
substr_replace(原字符串,要替代的字符串,开始替换的位置[,替换的长度])
```

其中,开始替换的位置从 0 开始计算,应小于原字符串的长度。要替换的长度是可

选的。

【例 9.11】 替换字符串。

```
<?php
echo substr_replace("abcdefgh", "DEF", 3);          // 将输出 "abcDEF"
echo substr_replace("abcdefgh", "DEF", 3, 2);       // 输出 "abcDEFfgh"
?>
```

以上只是介绍了 PHP 全部字符串操作函数中比较常用的一部分，更多的函数可以查看 PHP 手册。由于 PHP 是弱类型语言，所以其他类型的数据一般可以直接应用于字符串操作函数里，而自动转换成字符串类型，并进行处理，如下面的两个语句其结果是一样的。

```
echo substr("1234567", 1, 3);
echo substr(123456,1, 3);
```

9.5 使用数组

数组是一组有序的变量，其中每个变量叫做一个元素。数组中的每个元素都有自己的 ID，因此可以方便地访问它们。利用数组这种强大的数据类型，可以在 PHP 中实现复杂的程序逻辑。

9.5.1 数组的创建和显示

PHP 中的数组可以是一维数组，也可以是多维数组。创建数组使用的是 array()函数。语法格式如下：

```
array array([ $ keys = > ] $ values, … )
```

语法“ $ keys=> $ values”，用逗号分开，定义了关键字 $ keys 的键名和值 $ values，自定义键名可以是字符串或数字。如果省略了键名，会自动产生从 0 开始的整数作为键名，也可以自定义键名。

【例 9.12】 创建数组。

```
<?php
 $ array1 = array(1,2,3,4);
Print_r( $ array1);
//输出 Array([0] = > 1[1] = > 2[2] = > 3[3] = > 4)
 $ array2 = array("num" = 01,"name" = "张三","gender" = "男");
Print_r( $ array1);
//输出 Array([num] = > 01[name] = >张三[gender] = >男
 $ array3 = array(1 = > 2,2 = > 4,5 = > 6,8,10);
echo  $ array3[7];
//输出 10
?>
```

在例 9.12 中，数组 $ array1 的键名是自动产生的从 0 开始的整数，键名分别是 0、1、2、3。数组 $ array2 的键名是“num”、“name”、“gender”。数组 $ array3 中 8 和 10 的键名省

略,数组自动会给其分配键名,前一个数值 6 的键名是 5,所以 8 的键名是 6,10 的键名是 7。

打印数组函数 print_r($array),则会按照一定格式显示键名和值。

9.5.2 数组的分类

PHP 中主要有以下三种类型的数组。

1. 数值数组

数值数组存储的每个元素都带有一个数字 ID 键。

【例 9.13】 数值数组。

```
<?php
 $names[0] = "张三";
 $names[1] = "李四";
 $names[2] = "王五";
//开始输出
echo $names[1] . "和" . $names[2] . "是". $names[0] . "的好朋友";
//输出结果 李四和张三是王五的好朋友
?>
```

2. 关联数组

关联数组,它的每个 ID 键都关联一个值。在存储有关具体命名的值的数据时,使用数值数组不是最好的做法。通过关联数组,可以把值作为键,并向它们赋值。

【例 9.14】 关联数组。

在本例中,使用一个数组把年龄分配给不同的人:

```
 $ages = array("张三"=>32, "李四"=>30, "王五"=>34);
```

也可以在脚本中使用 ID 键:

```
<?php
 $ages['张三'] = "32";
 $ages['李四'] = "30";
 $ages['王五'] = "34";
echo "张三" . $ages['张三'] . "岁了";
?>
```

以上代码会输出下面的结果:

```
张三 32 岁了
```

3. 多维数组

在多维数组中,主数组中的每个元素也是一个数组。在子数组中的每个元素也可以是数组,以此类推。

在例 9.15 中,创建了一个带有自动分配的 ID 键的多维数组。

【例 9.15】　多维数组。

```
$families = array
(
"张三"=>array
  (
"张小文",
"张小亮",
"张小明"
  ),
"李四"=>array
  ("李小明"),
"王五"=>array
  (
"王小文",
"王小亮",
"王小明"
  )
);
//输出数组中的内容
echo $families['张三'][2];
```

以上代码会输出下面的结果：

```
张小明
```

9.6 流程控制

流程控制就是控制程序的执行方向，程序中通过流程控制语句来实现。PHP 程序中的流程控制语句主要包括条件控制语句、循环控制语句，而且控制语句可以嵌套来使用。

9.6.1 条件控制

PHP 提供了条件语句来执行基于不同条件的不同动作，进而实现程序的逻辑控制。常用的条件语句包括 if、elseif、else 以及 switch，其中，switch 是基于多个条件，其他的则是基于单个条件。

1. if…else

如果希望在某个条件成立时执行一些代码，在条件不成立时执行另一些代码，使用 if…else 语句。

语法格式如下：

```
if (condition)
  code to be executed if condition is true;
else
  code to be executed if condition is false;
```

【例 9.16】 if…else 程序示例。

如果当前日期是周五,输出“周末快乐!”,否则会输出“天天快乐!”:

```
<html>
<body>
<?php
 $d=date("D");
if ($d=="Fri")
  echo "周末快乐!";
else
  echo "天天快乐!";
?>
</body>
</html>
```

2. elseIf

如果希望在多个条件之一成立时执行代码,使用 elseif 语句。

语法格式如下:

```
if (condition)
  code to be executed if condition is true;
elseif (condition)
  code to be executed if condition is true;
else
  code to be executed if condition is false;
```

【例 9.17】 elseif 语句示例。

如果当前日期是周五,输出“周末快乐!”,如果是周日,则输出“周日快乐!”,否则输出“天天快乐!”:

```
<html>
<body>
<?php
 $d=date("D");
if ($d=="Fri")
  echo "周末快乐!";
elseif ($d=="Sun")
  echo "周日快乐!";
else
  echo "天天快乐!";
?>
</body>
</html>
```

3. switch

如果希望有选择地执行若干代码块之一,使用 switch 语句。

使用 switch 语句可以避免冗长的 if…elseif…else 代码块。

语法格式如下：

```
switch (expression)
{
case label1:
  code to be executed if expression = label1;
  break;
case label2:
  code to be executed if expression = label2;
  break;
default:
  code to be executed
  if expression is different
  from both label1 and label2;
}
```

9.6.2 循环控制

在编写代码时，经常需要让相同的代码块运行很多次，这时可以在代码中使用循环语句来完成这个任务。在 PHP 中，可以使用下列循环语句。

1. while

只要指定的条件成立，则循环执行代码块。

语法格式如下：

```
while (condition)
    code to be executed;
```

【例 9.18】 while 语句示例。

只要变量 i 小于或等于 5，代码就会一直循环执行下去。

```
<html>
<body>
<?php
$i=1;
//while 循环
while($i<=5)
{
  echo "数字: " . $i . "<br/>";
  $i++;
}
?>
</body>
</html>
```

2. do…while

首先执行一次代码块，然后在指定的条件成立时重复这个循环。

语法格式如下：

```
do
{
code to be executed;
}
while (condition);
```

【例 9.19】 do…while 程序示例。

下面的例子将对 i 的值进行一次累加，然后，只要 i 小于 5 的条件成立，就会继续累加下去。

```
<html>
<body>
<?php
 $i=0;
//do...while 循环
Do
{
   $i++;
  echo "数字: " . $i . "<br />";
}
while ($i<5);
?>
</body>
</html>
```

3. for

循环执行代码块指定的次数。

语法格式如下：

```
for (initialization; condition; increment)
{
  code to be executed;
}
```

注释：for 语句有三个参数。第一个参数初始化变量，第二个参数保存条件，第三个参数包含执行循环所需的增量。如果 initialization 或 increment 参数中包括多个变量，需要用逗号进行分隔。而条件必须计算为 true 或者 false。

【例 9.20】 for 语句示例。

下面的例子会把文本“Hello World!”显示 5 次。

```
<html>
<body>
<?php
//for 循环
for ($i=1; $i<=5; $i++)
{
  echo "Hello World!<br />";
```

```
}
?>
</body>
</html>
```

4. foreach

根据数组中每个元素来循环代码块。

每进行一次循环，当前数组元素的值就会被赋值给 value 变量(数组指针会逐一地移动)，以此类推。

语法格式如下：

```
foreach (array as value)
{
    code to be executed;
}
```

【例 9.21】 foreach 语句示例。

下面的例子示范了一个循环，这个循环可以输出给定数组的值。

```
<html>
<body>
<?php
//定义数组
$arr = array("一", "二", "三");
//foreach 循环遍历
foreach ($arr as $value)
{
  echo "值: " . $value . "<br />";
}
?>
</body>
</html>
```

9.7 使用函数使代码模块化

函数是一种可以在任何被需要的时候执行的代码块。在 PHP 中同样可以自己来定义函数。通过自定义函数可以使程序代码结构清晰易懂。通过这种模块化的程序结构设计，可以很大程度地减少开发工作量，从而提高代码编写的工作效率。

1. 创建函数

PHP 提供了自定义函数的功能，定义函数的格式如下：

```
function 函数名 (参数列表)
{
函数功能;
}
```

PHP 中的自定义函数需要遵守以下规范。

(1) 所有的函数都使用关键词 function()来开始。

(2) 函数的名称应该提示出它的功能。函数名称以字母或下划线开头。

(3) 开口的花括号之后的部分是函数的代码。

(4) 插入函数代码。

(5) 函数通过关闭花括号来结束。

【例 9.22】 自定义函数示例。

注意：函数名对大小写敏感。

```
<html>
<body>
<?php
//定义函数
function writeMyName()
{
  echo "张三";
}
?>
</body>
</html>
```

2. 使用函数

创建完函数后就可以在 PHP 脚本中使用这个函数了。

【例 9.23】 使用函数示例。

```
<html>
<body>
<?php
//定义函数
function writeMyName()
{
  echo "张三";
}
echo "大家好!<br />";
echo "我的名字是";
//调用函数
writeMyName();
echo ".<br />";
//调用函数
writeMyName();
echo "是我的名字.";
?>
</body>
</html>
```

以上代码的输出如下：

```
大家好!
```

我的名字是张三。
张三是我的名字。

3. 返回值

函数声明时，在函数代码中使用 return 语句可以结束函数的运行，程序返回到调用该函数的下一条语句。

【例 9.24】 return 语句示例。

```
<html>
<body>
<?php
//定义函数支持返回值
function add( $x, $y)
{
   $total = $x + $y;
  return $total;
}
//调用函数
echo "1 + 16 = " . add(1,16);
?>
</body>
</html>
```

以上代码的输出如下：

```
1 + 16 = 17
```

小结

本章系统地介绍了 PHP 语言的基础语法，通过这一章的学习，可以对 PHP 语言有个概要性的理解。事实上，PHP 在语法上与 C++ 特别相似，对于有一定 C 语言学习基础的读者来说，可以很快掌握。而且 PHP 语言在某些语法特性上还进行了简化，相比其他编程语言来说相对简单，更加快速灵活，这也是它的优势所在。

习题

1. 变量如何定义？
2. PHP 中变量有哪些基本数据类型？
3. 计算字符串长度用到哪个函数？
4. 定义数组用到哪个函数？
5. 定义一个函数，用于比较两个数的大小，如果前者比后者大则返回 1，反之返回 0。

第10章 PHP处理表单数据

本章要点：

- HTML 表单；
- 动态生成表单；
- 提交和接收表单数据；
- 表单数据验证；
- 通过表单上传文件。

在本章中，将学习如何通过 PHP 来获取表单中的用户输入并执行相应的处理。

10.1 HTML 表单

表单是 Web 应用中最常用的组件，由“提交”按钮以及其他相关元素组成。表单被应用在各个领域中，可以实现注册用户、填写信息和登录、查询等功能。因此，创建和处理表单是 PHP 开发者必须要掌握的一项关键技术。

在学习具体的 PHP 处理表单的技术之前，要掌握使用 HTML 如何创建表单。一个 HTML 表单，实际上是由一系列被嵌入到 Web 页面中的简单 HTML 标签所组成的。这些元素包括：文本输入框、复选框、单选框和下拉框等。

所有的 Web 表单元素被嵌入在<form></form>标签中。

【例 10.1】 表单标签。

```
< form action = "myscript.php" method = "post">
<! -- 表单内容 -- >
< /form >
```

常用的 HTML 表单元素如表 10.1 所示。

表 10.1 PHP 常用表单元素

表 单 元 素	说 明
input type="checkbox"	复选框，允许用户选择多个选择项
input type="file"	文件浏览框，当文件上传时，可用来打开一个模式窗口以选择文件
input type="hidden"	隐藏标签，它不会在用户浏览的页面界面上出现，当用户填写资料表单和跨页之间传值时，可以使用该标签传递一些隐含的值

续表

表单元素	说明
input type="password"	密码文本框,用户在该文本框输入字符时将被替换显示为 * 号,但是密码并没有加密,只是被 * 替换显示,通过查看 HTML 源代码还是可以看到原始值内容
input type="radio"	单选项,用于设置一组选择项,用户只能选择一个
input type="reset"	清除与重置表单内容,用于清除表单中所有文本框的内容,而且使选择菜单项恢复到初始值
input type="submit"	表单提交按钮

在表单标签<form>中提供了以下两个内置属性。

(1) action 属性:此属性告诉浏览器将用户输入的表单数据提交到哪个地址去处理。这个地址必须是一个标准的 URL 绝对路径(例如 http://www.example.com/myscript.php)或者一个相对路径(例如 myscript.php,/myscript.php 或者../scripts/myscript.php)。该路径对应的脚本会对接收到的数据进行下一步的处理。

(2) method 属性:此属性告诉浏览器以何种方式提交数据,分别是 GET 和 POST 方式,如果没有设置 method 属性或该属性为空值,浏览器默认 method 的值为 POST 方法。

GET 方式是在访问 URL 时,使用浏览器地址栏来传递值,这种方式简单直观,适合发送少量的数据,但缺点是访问该网站的用户也可以修改 URL 串后发送给服务器,如果程序处理得不够好很容易出错,而且 GET 传递的字符串长度不能超过 250 个字符,如果超长,浏览器会自动截断,导致数据缺失。

而 POST 方式则是随 HTTP 的<header>中的信息一起发送表单信息,用户不能随意修改,这对于 Web 应用程序而言,安全性要好得多,并且 POST 方式也能够支持大数据量用户输入数据的发送。

使用 POST 方式提交表单,理论上数据的大小无上限。但在实际使用中,如果使用 PHP 进行 POST 提交时,文件大小会受 PHP 配置文件(php.ini)限制,默认是 2MB。可以修改 php.ini 文件中的 post_max_size 参数,将其修改为自己需要的大小,但由于 HTTP 的特性,这个值不宜设置过大,最大以 8MB 为宜。

在开发中需要根据实际应用灵活选择 GET 和 POST 来提交表单数据。

【例 10.2】 HTML 表单的源代码示例。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>基本的 HTML 表单</title>
</head>
<body>
<h1>An HTML Form</h1>
<form action="test.php" method="get">
<div style="width: 25em;">
<label for="textField">A text input field</label>
<input type="text" name="textField" id="textField" value="" />
<label for="passwordField">A password field</label>
```

```
< input type = "password" name = "passwordField" id = "passwordField"
value = "" />
< label for = "checkboxField"> A checkbox field </label >
< input type = "checkbox" name = "checkboxField" id = "checkboxField"
value = "yes" />
< label for = "radioButtonField1"> A radio button field </label >
< input type = "radio" name = "radioButtonField" id = "radioButtonField1"
value = "radio1" />
< label for = "radioButtonField2"> Another radio button </label >
< input type = "radio" name = "radioButtonField" id = "radioButtonField2"
value = "radio2" />
< label for = "submitButton"> A submit button </label >
< input type = "submit" name = "submitButton" id = "submitButton"
value = "Submit Form" />
< label for = "resetButton"> A reset button </label >
< input type = "reset" name = "resetButton" id = "resetButton"
value = "Reset Form" />
< label for = "fileSelectField"> A file select field </label >
< input type = "file" name = "fileSelectField" id = "fileSelectField"
value = "" />
< label for = "hiddenField"> A hidden field </label >
< input type = "hidden" name = "hiddenField" id = "hiddenField" value = "" />
< label for = "imageField"> An image field </label >
< input type = "image" name = "imageField" id = "imageField" value = ""
src = "asterisk.gif" width = "23" height = "23" />
< label for = "pushButton"> A push button </label >
< input type = "button" name = "pushButton" id = "pushButton"
value = "Click Me" />
< label for = "pullDownMenu"> A pull – down menu </label >
< select name = "pullDownMenu" id = "pullDownMenu" size = "1">
< option value = "option1"> Option 1 </option >
< option value = "option2"> Option 2 </option >
< option value = "option3"> Option 3 </option >
</select >
< label for = "listBox"> A list box </label >
< select name = "listBox" id = "listBox" size = "3">
< option value = "option1"> Option 1 </option >
< option value = "option2"> Option 2 </option >
< option value = "option3"> Option 3 </option >
</select >
< label for = "multiListBox"> A multi – select list box </label >
< select name = "multiListBox" id = "multiListBox" size = "3"
multiple = "multiple">
< option value = "option1"> Option 1 </option >
< option value = "option2"> Option 2 </option >
< option value = "option3"> Option 3 </option >
</select >
< label for = "textAreaField"> A text area field </label >
< textarea name = "textAreaField" id = "textAreaField" rows = "4"
cols = "50"></textarea >
< input type = "submit" />
```

```
</div>
</form>
</body>
</html>
```

上面的 HTML 页面实例包含一系列输入框和一个“提交”按钮。当用户填写该表单并单击“提交”按钮时，表单的数据会被送往 test.php 这个文件。将源代码保存为一份 HTML 文件并使用浏览器打开，就可以看到结果，最终结果如图 10.1 所示。

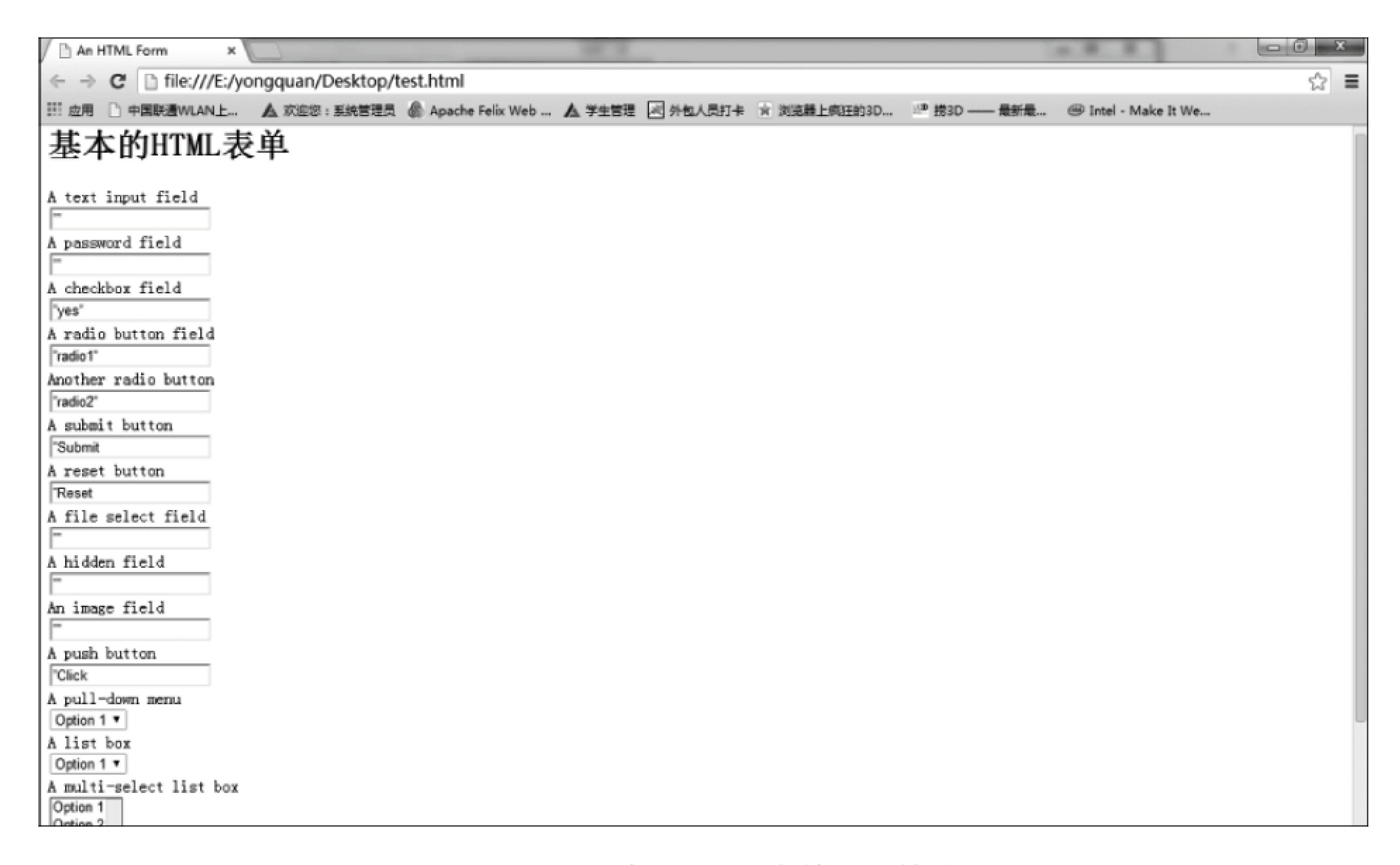


图 10.1　基本 HTML 表单显示效果

10.2　动态生成表单

在很多场景中，表单并不是固定不变的，而是需要根据特定情况动态修改表单中的 HTML 元素，例如，购买火车票时进行查询的表单，这就是一种需要动态更新内容的表单，它需要从数据库中查询出有效的车次等信息供用户进行选择。如果有一个动态生成表单及其验证脚本的程序，就可以大大减少制作这些表单的工作量。PHP 作为一种动态语言，可以很好地实现这一需求。

一般来说，可以通过以下两种方式来动态生成表单。

(1) 直接输出：在 PHP 代码中通过调用 echo 或者 print 方法将对应 Web 表单元素的 HTML 源代码支持输出。

(2) 嵌入输出：可以通过＜? php…? ＞标签的方式将 HTML 源代码嵌入到 PHP 片段中。

当然，也可以混合使用上述两种方式来生成表单。

【例 10.3】 嵌入了 PHP 脚本的动态表单页面示例。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>注册表单</title>
<link rel="stylesheet" type="text/css" href="common.css" />
<style type="text/css">
.error { background: #d33; color: white; padding: 0.2em; }
</style>
</head>
<body>
<?php
if ( isset( $_POST["submitButton"] ) ) {
    processForm();
} else {
    displayForm( array() );
}
function validateField( $fieldName, $missingFields ) {
    if ( in_array( $fieldName, $missingFields ) ) {
       echo 'class="error"';
    }
}
function setValue( $fieldName ) {
    if ( isset( $_POST[$fieldName] ) ) {
        echo $_POST[$fieldName];
    }
}
function setChecked( $fieldName, $fieldValue ) {
    if ( isset( $_POST[$fieldName] ) and $_POST[$fieldName] == $fieldValue ) {
        echo 'checked="checked"';
    }
}
function setSelected( $fieldName, $fieldValue ) {
    if ( isset( $_POST[$fieldName] ) and $_POST[$fieldName] == $fieldValue ) {
        echo 'selected="selected"';
    }
}
//自定义表单处理函数
function processForm() {
    $requiredFields = array( "firstName", "lastName", "password1",
    "password2", "gender" );
    $missingFields = array();
    foreach ( $requiredFields as $requiredField ) {
        if ( !isset( $_POST[$requiredField] ) or !$_POST[$requiredField] ) {
            $missingFields[] = $requiredField;
        }
    }
    if ( $missingFields ) {
        displayForm( $missingFields );
```

```
    } else {
        displayThanks();
    }
}
function displayForm( $missingFields ) {
?>
    <h1>注册表单</h1>
<?php if ( $missingFields ) { ?>
    <p class="error">发现错误</p>
<?php } else { ?>
    <p>感谢注册.</p>
<?php } ?>
<form action="registration.php" method="post">
<div style="width: 30em;">
<label for="firstName"<?php validateField( "firstName",
$missingFields ) ?>>姓 *</label>
<input type="text" name="firstName" id="firstName"
value="<?php setValue( "firstName" ) ?>" />
<label for="lastName"<?php validateField( "lastName",
$missingFields ) ?>>名 *</label>
<input type="text" name="lastName" id="lastName" value=
"<?php setValue( "lastName" ) ?>" />
<label for="password1"<?php if ( $missingFields ) echo
'class="error"'?>>输入密码 *</label>
<input type="password" name="password1" id="password1" value="" />
<label for="password2"<?php if ( $missingFields ) echo
"F"<?php setChecked( "gender", "F" )?> />
<label for="favoriteWidget">最喜欢的栏目? *</label>
<select name="widget1" id="favoriteWidget" size="1">
<option value="superWidget"<?php setSelected( "favoriteWidget",
"widget1" ) ?>>栏目 1</option>
<option value="widget2"<?php setSelected( "favoriteWidget",
"widget2" ) ?>>栏目 2</option>
<option value="widget3"<?php setSelected( "favoriteWidget",
"widget3" ) ?>>栏目 3</option>
</select>
<label for="newsletter">是否愿意接收电子邮件?
</label>
<input type="checkbox" name="newsletter" id="newsletter" value="yes"
<?php setChecked( "newsletter", "yes" ) ?> />
<label for="comments">建议?</label>
<textarea name="comments" id="comments" rows="4" cols="50"><?php
setValue( "comments" ) ?></textarea>
<div style="clear: both;">
<input type="submit" name="submitButton" id="submitButton" value=
"提交" />
<input type="reset" name="resetButton" id="resetButton"
value="Reset Form" style="margin-right: 20px;" />
```

```
</div>
</div>
</form>
<?php
}
function displayThanks() {
?>
    <h1>谢谢</h1>
    <p>您的申请已收到.</p>
<?php
}
?>
</body>
</html>
```

可以看到,动态生成表单本质上是通过 PHP 代码对 HTML 代码进行拼接处理,最终生成用户看到的 HTML 表单的过程。

10.3 提交和接收表单数据

用户在表单中输入数据后,只需单击“提交”按钮,就可以将信息提交到服务器端。此时,就可以通过 PHP 脚本来对用户提交的数据进行处理。在 PHP 中处理表单提交的数据非常简单,它能够自动地将由客户端表单发送的数据赋值给相应变量,从而极大地简化了整个表单的处理过程。

根据表单提交数据所使用的方法(method)不同,需要通过不同的变量获取内容。如果使用了 GET 方法,那么表单中的信息将被储存到全局变量 $_GET 中,如果使用了 POST 方法,则相关信息会被存储到全局变量 $_POST 中。如果并不关心请求数据的来源,也可以用全局变量 $_REQUEST,它包含所有 GET、POST、COOKIE 和 FILE 的数据。

上述三个全局变量都是以 Key-Value 数组的形式提供,一般将存放用户输入内容的 HTML 表单控件的编号作为索引。

例 10.4 和例 10.5 是一个简单的用户注册的示例,这段脚本将获取用户输入的信息并将其显示在一个结果页面中。这个例子由两部分组成,registration.html 供用户输入注册信息并将相关数据提交给服务器,process.php 则负责接收数据并将其格式化后显示给用户。

【例 10.4】 registration.html 页面代码。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Membership Form</title><link rel="stylesheet" type="text/
css" href="common.css" />
```

```
</head>
<body>
<h1>注册页面</h1>
<p>请输入以下信息完成注册</p>
<form action="process_registration.php" method="post">
<div style="width: 30em;">
<label for="firstName">姓</label>
<input type="text" name="firstName" id="firstName" value="" />
    <label for="lastName">名</label>
    <input type="text" name="lastName" id="lastName" value="" />
    <label for="password1">密码</label>
    <input type="password" name="password1" id="password1" value="" />
    <label for="password2">重新输入密码</label>
    <input type="password" name="password2" id="password2" value="" />
    <label for="genderMale">您的性别...</label>
    <input type="radio" name="gender" id="genderMale" value="M" />
    <label for="genderFemale">...or female?</label>
    <input type="radio" name="gender" id="genderFemale" value="F" />
    <label for="favoriteWidget">你的爱好?</label>
    <select name="favoriteWidget" id="favoriteWidget" size="1">
    <option value="superWidget">游泳</option>
    <option value="megaWidget">旅游</option>
    <option value="wonderWidget">看书</option>
    </select>
    <label for="newsletter">是否愿意接收电子邮件推广信息?</label>
    <input type="checkbox" name="newsletter" id="newsletter" value="yes" />
    <label for="comments">其他备注?</label>
    <textarea name="comments" id="comments" rows="4"
    cols="50"></textarea>
    <div style="clear: both;">
<input type="submit" name="submitButton" id="submitButton"value=" Send Details" />
    <input type="reset" name="resetButton" id="resetButton"
    value="Reset Form" style="margin-right: 20px;" />
    </div>
    </div>
    </form>
    </body>
</html>
```

【例 10.5】 process.php 页面代码。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>注册成功</title>
```

```
<link rel="stylesheet" type="text/css" href="common.css" />
</head>
<body>
<h1>谢谢</h1>
<p>注册成功. 注册信息如下:</p>
<dl>
<dt>姓</dt><dd><?php echo $_POST["firstName"]?></dd>
<dt>名</dt><dd><?php echo $_POST["lastName"]?></dd>
<dt>密码</dt><dd><?php echo $_POST["password1"]?></dd>
<dt>性别</dt><dd><?php echo $_POST["gender"]?></dd>
<dt>爱好</dt><dd><?php echo $_POST["favoriteWidget"]?></dd>
<dt>是否愿意注册电子邮件推广信息?</dt><dd><?php echo
 $_POST["newsletter"]?></dd>
<dt>备注</dt><dd><?php echo $_POST["comments"]?></dd>
</dl>
</body>
</html>
```

该表单包括常用表单元素：单行文本框、多行文本框、单选项(radio)、多选项(checkbox)，以及多选菜单。下面进行详细的说明。

(1) maxlength 是与密码文本框关联的属性，它限制用户输入密码的最大长度为 10 个字符。

(2) favoriteWidget 列表框是列表菜单，它由一组选项组成。通过 selected 属性来标记哪个选项被选中了。

(3) comments 文本框中的内容，按照 rows 和 cols 显示文字、行和列宽。

(4) newsletter 是一个复选按钮(checkbox)，和单选项一样，所有多选项成员也需有同名的属性，且属性名称需要添加括号“[]”，这样就把多选项的值以数组形式发送给 PHP。

(5) checked 标签是指单选项和多选项中的某个值，默认已经被选择。

运行的结果如图 10.2 和图 10.3 所示。

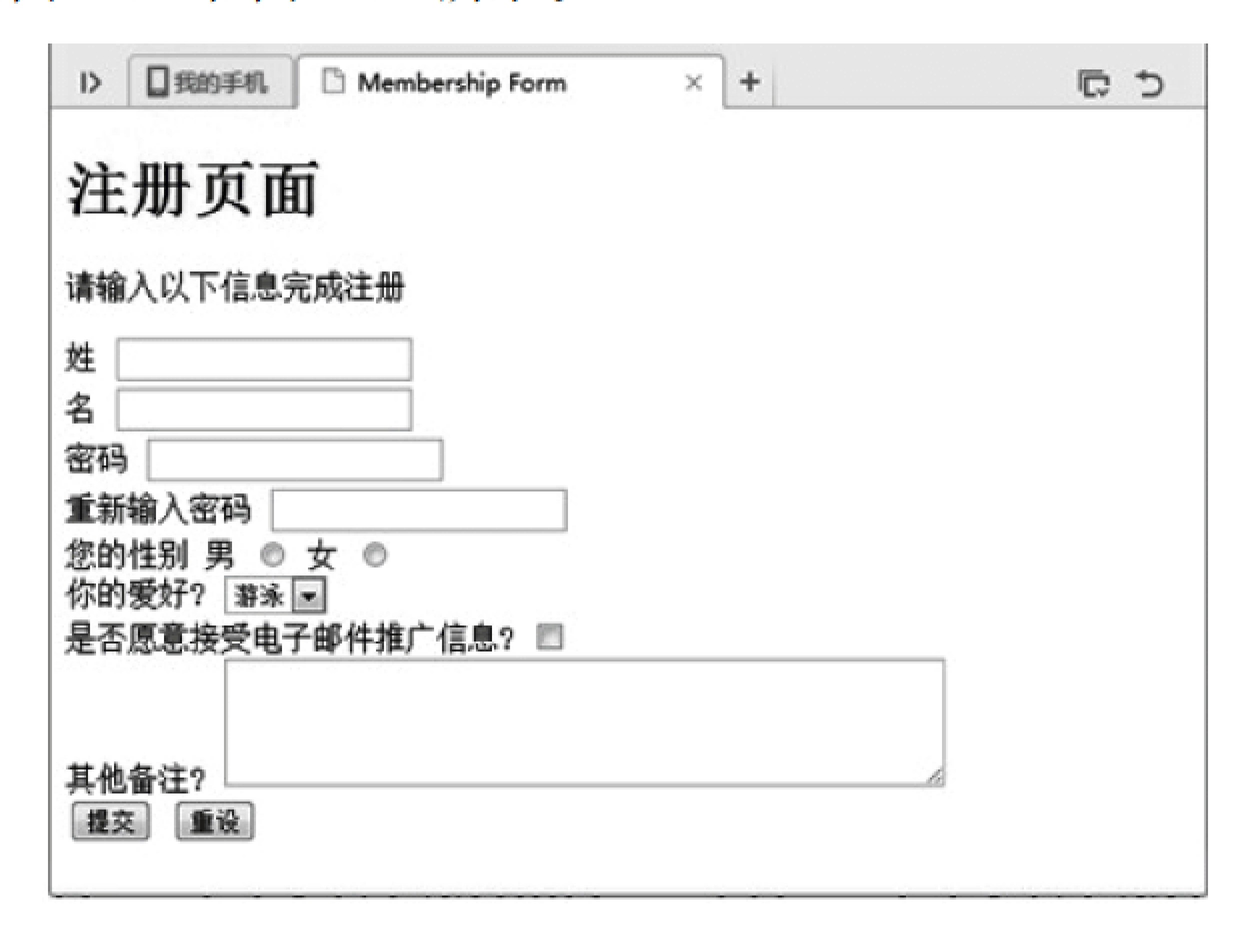


图 10.2　表单示例运行效果

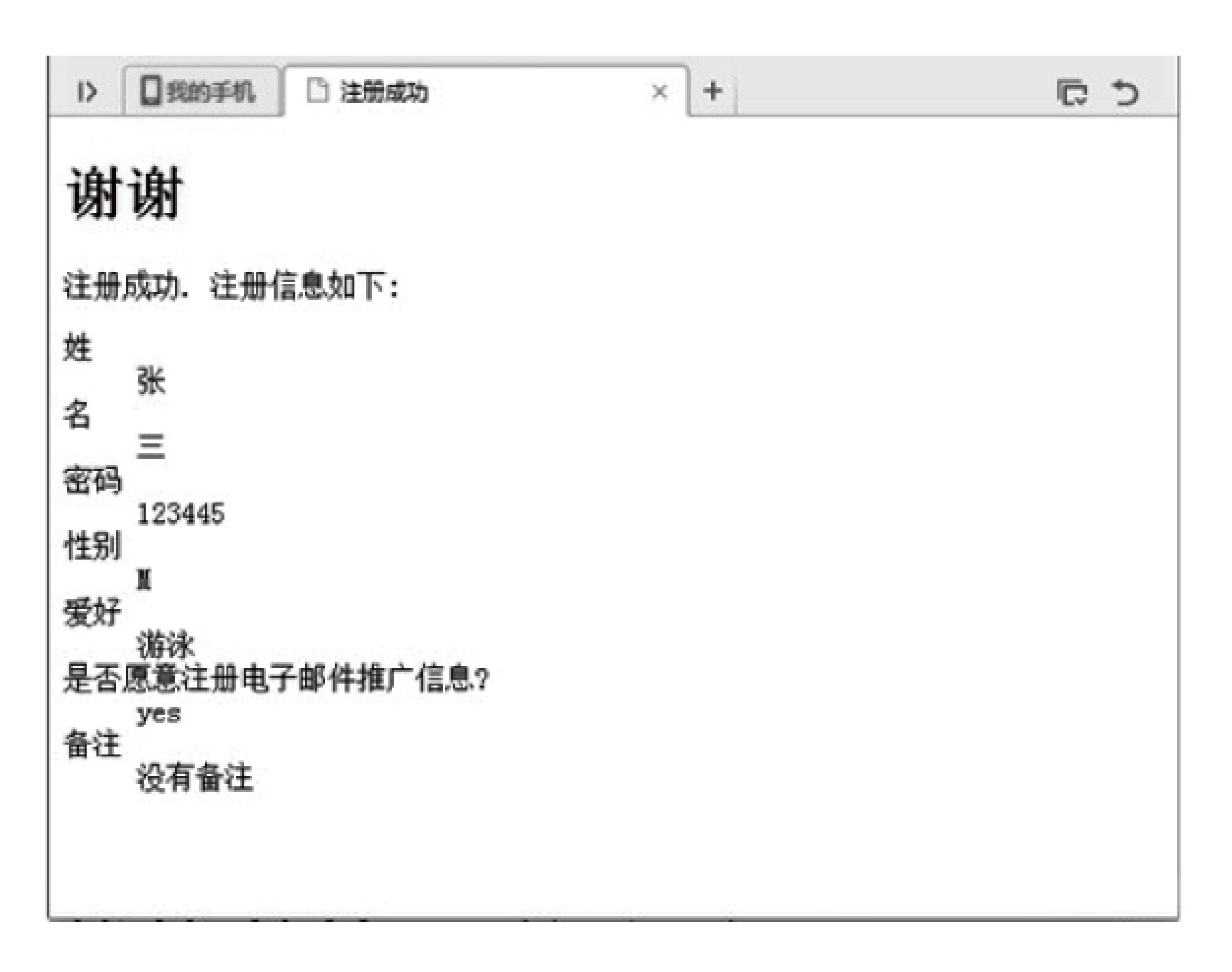


图 10.3 表单接收数据效果

10.4 表单数据验证

验证表单数据是 Web 应用程序最重要的部分之一,在开发网站应用时,如果不进行验证,各种奇怪的错误或就会出现在程序员最意想不到的地方。通常为了避免这种情况,开发者会给表单加一个验证的过程,在表单数据提交到服务器之前或者服务器之后对用户填写的数据(信息的长度、格式、内容等)进行验证,若遇到错误的填写则终止处理并返回要求用户进行更正。

10.4.1 表单数据验证方式

表单数据的验证分为两种:服务器端校验和客户端校验。客户端校验实际上就是包含在已下载的页面中的一段 JS 脚本,当用户提交表单时,通过运行本地 Script 来进行验证,客户端的验证速度很快,并且可以减轻服务器的负载。而服务器端的验证则是将页面提交到服务器处理,服务器上的另一个 PHP 页面先执行对表单的验证,然后再返回结果到客户端。

如果表单访问的是数据库,就非常有必要采用服务器端的验证。一般服务器端验证页面会将表单传给它自己,而不是跳转到不同的页面。这样用户就可以在同一张表单页面得到错误信息,用户也就更容易发现错误了。服务器端校验的缺点是每一次验证都要经过服务器,消耗时间较长,但对于关键业务逻辑必须要通过服务器段验证来保证数据安全性。

在本节中主要介绍如何使用 PHP 在服务器端进行输入验证。比如,在用户注册时,要求用户的输入必须是中文(英文或数字);要求用户的输入必须是有效的邮件地址;对用户输入的数据进行各种不同的限制;限制用户输入的数据量等。

根据不同的应用场景,需要进行一些特殊的校验。针对这些需要进行校验的地方,可以采取的方法主要包括以下几种。

(1) 使用 htmlspecialchars()函数将 HTML 特殊字符进行过滤;
(2) 使用函数 trim()去除数据的前后空格;
(3) 使用函数 strlen()判断数据的长度;
(4) 使用函数 is_numeric()判断数据是否纯数字;
(5) 使用正则表达式来验证数据是否满足特定规则。

10.4.2 正则表达式验证表单

正则表达式是程序开发中一个重要的元素,它可以用来描述或匹配文本的字符串,如特定的字符、词或算式等。虽然编写一个特定的正则表达式比较复杂和费时,但其执行效率却很高,尤其适合用于进行快速数据验证。下面的例子列出了常见的用于数据验证的几种正则表达式。

【例 10.6】 验证 E-mail 地址。

```
<?php
if (!isset($_POST['send']) || $_POST['send']!= '提交') {
header('Location:Demo1.php');
exit;
}
if (preg_match('/([\w\.]{2,255})@([\w\-]{1,255}).([a-z]{2,4})/', $_POST['email'])) {
echo '电子邮件合法';
} else {
echo '电子邮件不合法';
}
?>
```

【例 10.7】 验证用户名。

```
<?php
$username = "user_name12";
if (preg_match('/^[a-z\d_]{5,20}$/i', $username)) {
    echo "用户名合法.";
} else {
    echo "用户名不合法";
}
?>
```

这是一个用于验证用户名的实例,其中包括字母、数字(A～Z,a～z,0～9)、下划线以及最低 5 个字符,最大 20 个字符的限制。同时,也可以根据需要,对最小值和最大值做合理的修改。

【例 10.8】 验证电话号码。

```
<?php
$phone = "(021)423-2323";
if (preg_match('/\(?\d{3}\)?[-\s.]?\d{3}[-\s.]\d{4}/x', $phone)) {
    echo "电话号码合法";
} else {
    echo "电话号码不合法";
```

```
}
?>
```

【例 10.9】 验证 IP 地址。

```
<?php
$IP = "198.168.1.78";
if (preg_match('/^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5]).){3}([1-9]?[0-9]|
1[0-9]{2}|2[0-4][0-9]|25[0-5])$/', $IP)) {
    echo " IP 地址合法.";
} else {
    echo "IP 地址不合法";
}?>
```

【例 10.10】 验证生日。

```
<?php
$ birthday = "2008-06-19";
if(!preg_match("/^(19|20)\d{2}-(0?\d|1[012])-(0?\d|[12]\d|3[01])$/", $birthday))
{
    $erro_msg. = $st_birthday."日期格式错误!正确日期格式为 yyyy-mm-dd! ";
    echo $erro_msg;
}
Else
{
    echo "日期正确"
}
?>
```

验证生日是否为合理格式,如果不正确提示为 yyyy-mm-dd 的格式类型。

【例 10.11】 验证邮政编码。

```
<?php
$zipcode = "12345-5434";
if (preg_match("/^([0-9]{5})(-[0-9]{4})?$/i", $zipcode)) {
    echo "邮政编码合法";
} else {
    echo "邮政编码不合法";
}
?>
```

【例 10.12】 验证信用卡号。

```
<?php
$cc = "378282246310005";
if (preg_match('/^(?:4[0-9]{12}(?:[0-9]{3})?|5[1-5][0-9]{14}|6011[0-9]{12}|3(?:0[0-
5]|[68][0-9])[0-9]{11}|3[47][0-9]{13})$/', $cc)) {
    echo "信用卡号合法";
} else {
    echo "信用卡号不合法";
}
?>
```

【例 10.13】 验证域名。

```
<?php
$url = "http://www.nuc.edu/";
if (preg_match('/^(http|https|ftp):\/\/([A-Z0-9][A-Z0-9_-]*(?:\.[A-Z0-9][A-Z0-9_-]*)+):?(\d+)?\/?/i', $url)) {
    echo "域名合法";
} else {
    echo "域名不合法";
}
?>
```

【例 10.14】 UTF8 下的中文验证。

```
<?php
if (preg_match("/^[".chr(0xa1)."-".chr(0xff)."]+$/", $st_name){
    print("该字符串全部是中文");
} else {
   print("该字符串不全部是中文");
}
?>
```

【例 10.15】 GB2312 下中文的验证。

```
<?php
$str1 = "编程";
$str = "php 编程";
if (preg_match("/^[\x{4e00}-\x{9fa5}]+$/u", $str)) {
   print("该字符串全部是中文");
} else {
   print("该字符串不全部是中文");
}
?>
```

10.4.3 过滤式扩展验证表单

除了使用正则表达式进行输入验证外，PHP 还提供了过滤器扩展的方式来进行输入验证，PHP 过滤器主要用于验证和过滤来自非安全来源的数据，这种方式相比正则表达式更简单，使得数据过滤更轻松快捷。

下面就是 PHP 提供的用于过滤变量的过滤器函数。

(1) filter_var()：通过一个指定的过滤器来过滤单一的变量。

(2) filter_var_array()：通过相同的或不同的过滤器来过滤多个变量。

(3) filter_input：获取一个输入变量，并对它进行过滤。

(4) filter_input_array：获取多个输入变量，并通过相同的或不同的过滤器对它们进行过滤。

【例 10.16】 使用 filter_var()函数验证一个整数。

```
<?php
$int = 123;
```

```
if(!filter_var( $ int, FILTER_VALIDATE_INT))
{
    echo("Integer is not valid");
}
else
{
    echo("Integer is valid");
}
?>
```

上面的代码使用了“FILTER_VALIDATE_INT”过滤器来过滤变量。由于这个整数是合法的，因此代码的输出是：“Integer is valid”。

假如尝试使用一个非整数的变量，则输出是：“Integer is not valid”。

1. Validating 和 Sanitizing

有两种过滤器，Validating 过滤器和 Sanitizing 过滤器。Validating 过滤器用于验证用户输入，判断其是否符合严格的格式规则（比如 URL 或 E-mail 验证），如果符合则返回预期的类型，否则返回 FALSE。Sanitizing 过滤器则用于允许或禁止字符串中指定的字符，它没有数据格式规则，并且始终返回字符串。

2. 选项和标志

选项和标志用于向指定的过滤器添加额外的过滤选项。不同的过滤器有不同的选项和标志。在 PHP 官方网站提供的 PHP Filter 参考手册中可以查询到完整的函数和过滤器列表以及每个过滤器的可用选项和标志。

【例 10.17】 使用 filter_var()和 min_range 以及 max_range 选项验证一个整数。

```
<?php
 $ var = 300;
 $ int_options = array(
    "options" = > array
    (
        "min_range" = > 0,
        "max_range" = > 256
    )
);
if(!filter_var( $ var, FILTER_VALIDATE_INT, $ int_options))
{
    echo("不合法的数字");
}
else
{
    echo("合法的数字");
}
?>
```

就像上面的代码一样，选项必须放入一个名为“options”的相关数组中。如果使用标志，则不需在数组内。由于整数是“300”，它不在指定的范围内，以上代码的输出将是“不合

法的数字”。

接下来，可以通过使用过滤器来验证来自表单的输入。需要做的第一件事情是确认是否存在之前正在查找的输入数据，然后用 filter_input()函数过滤输入的数据。

【例 10.18】 输入变量 email 被传到 PHP 页面。

```
<?php
if(!filter_has_var(INPUT_GET, "email"))
{
    echo("没有 email 字段");
}
else
{
    if (!filter_input(INPUT_GET, "email", FILTER_VALIDATE_EMAIL))
    {
        echo "E-Mail 地址错误";
    }
    else
    {
        echo "E-Mail 地址正确";
    }
}
?>
```

例 10.18 先检测其是否存在 GET 类型的 email 输入变量，如果存在输入变量，再检测它是否是有效的邮件地址，最后输出验证结果。

接下来，可以使用过滤器清理一下从表单传来的 URL。

首先，要确认是否存在之前正在查找的输入数据，然后，可以用 filter_input()函数来净化输入数据。

【例 10.19】 输入变量 url 被传到 PHP 页面。

```
<?php
if(!filter_has_var(INPUT_POST, "url"))
{
    echo("url 字段不存在");
}
else
{
    $url = filter_input(INPUT_POST, "url", FILTER_SANITIZE_URL);
}
?>
```

例 10.19 先检测是否存在 POST 类型的 url 输入变量，如果存在此输入变量，对其进行净化（删除非法字符），并将其存储在 $url 变量中。假如输入变量类似这样：“http://www.nuc#$%uc.edu.cn/”，则净化后的 $url 变量应该是这样的：“http://www.nuc.edu.cn/”。

在一个表单中，通常会存在多个输入字段。为了避免对 filter_var 或 filter_input 重复调用，可以使用 filter_var_array 或 filter_input_array 函数。

【例 10.20】 使用 filter_input_array()函数过滤三个 GET 变量。

```
<?php
$filters = array
(
    "name" => array
    (
        "filter" => FILTER_SANITIZE_STRING
    ),
    "age" => array
    (
        "filter" => FILTER_VALIDATE_INT,
        "options" => array
         (
            "min_range" => 1,
            "max_range" => 120
         )
     ),
    "email" => FILTER_VALIDATE_EMAIL,
);
$result = filter_input_array(INPUT_GET, $filters);
if (!$result["age"])
{
    echo("年龄必须是大于 1 小于 120 的数字.<br />");
}
elseif(!$result["email"])
{
    echo("E-Mail 地址不合法<br />");
}
else
{
    echo("用户输出错误");
}
?>
```

例 10.20 有三个通过 GET 方法传送的输入变量(name, age 和 email),可以设置一个数组,其中包含输入变量的名称,以及用于指定的输入变量的过滤器,然后调用 filter_input_array 函数,参数包括 GET 输入变量及刚才设置的数组,接着检测 $result 变量中的 age 和 email 变量是否有非法的输入,如果存在非法输入,就提示用户。

需要注意的是,filter_input_array()函数的第二个参数可以是数组或单一过滤器的 ID。如果该参数是单一过滤器的 ID,那么这个指定的过滤器会过滤输入数组中所有的值。而如果该参数是一个数组,那么此数组必须遵循下面的规则。

(1) 必须是一个关联数组,其中包含的输入变量是数组的键(比如 age 输入变量)。

(2) 此数组的值必须是过滤器的 ID,或者是规定了过滤器、标志以及选项的数组。

很多时候在进行一些特殊的验证时需要调用自定义函数,此时可以使用 FILTER_CALLBACK 过滤器,可以将自定义的函数作为一个过滤器来使用。这样,程序就拥有了数据过滤的完全控制权。这时还可以创建自己的自定义函数,也可以使用已有的 PHP 函数。

设置调用函数的方式与设置选项相同，只需要调用 filter_var()函数，将 FILTER_CALLBACK 过滤器以及包含函数的数组当作参数传入即可。

【例 10.21】 使用一个自定义的函数把所有"_"转换为空格。

```
<?php
//定义自定义函数
function convertSpace( $ string)
{
    return str_replace("_", "", $ string);
}
 $ string = "左_中_右";
//输出结果
echo filter_var( $ string, FILTER_CALLBACK,array("options" = >"convertSpace"));
?>
```

以上代码的结果如下：

```
左中右
```

10.5 通过表单上传文件

在学习前面章节的时候了解到 PHP 通过 POST 方法可以提交表单数据，事实上，POST 方法不仅可以提交字符串类型的表单数据，还可以传递二进制数据，可以利用 PHP 的这一功能，将文件上传到服务器。

1. 创建一个文件上传表单

【例 10.22】 创建一个上传文件的 HTML 表单。

```
< html >
< body >
< form action = "upload_file. php" method = "post"enctype = "multipart/form - data">
< label for = "file">文件名:</label >
< input type = "file" name = "file" id = "file" />
< br />
< input type = "submit" name = "submit" value = "提交" />
</form >
</body >
</html >
```

在例 10.22 中，有关此表单的信息如下。

(1) <form>标签的 enctype 属性规定了在提交表单时要使用哪种内容类型。在表单需要二进制数据时，比如文件内容，请使用 multipart/form-data。

(2) <input>标签的 type="file"属性规定了应该把输入作为文件来处理。举例来说，当在浏览器中预览时，会看到输入框旁边有一个"浏览"按钮。

注意： 允许用户上传文件会存在一些安全风险，例如，非法用户可能会上传 PHP 木马程序，因此只能允许可信的用户执行文件上传操作。

2. 创建上传脚本

【例 10.23】 上传文件。

```
<?php
//接收从表单中 POST 过来的文件数据并处理
if ( $_FILES["file"]["error"] > 0)
{
  echo "错误: " . $_FILES["file"]["error"] . "<br />";
}
else
{
  echo "上传: " . $_FILES["file"]["name"] . "<br />";
  echo "类型: " . $_FILES["file"]["type"] . "<br />";
  echo "大小: " . ( $_FILES["file"]["size"] / 1024) . " Kb<br />";
  echo "存储为: " . $_FILES["file"]["tmp_name"];
}
?>
```

通过使用 PHP 的全局数组 $_FILES,可以从客户计算机向远程服务器上传文件。其中第一个参数是表单的 input name,第二个下标可以是"name","type","size","tmp_name"或"error"。具体解释如下。

(1) $_FILES["file"]["name"]:被上传文件的名称。

(2) $_FILES["file"]["type"]:被上传文件的类型。

(3) $_FILES["file"]["size"]:被上传文件的大小,以字节计。

(4) $_FILES["file"]["tmp_name"]:存储在服务器的文件的临时副本的名称。

(5) $_FILES["file"]["error"]:在文件上传过程中发生的错误代码。

这个例子只是简单演示了文件上传,在实际项目中,应该增加有关哪类用户有权上传文件的限制,来增强网站的安全性。

3. 上传限制

【例 10.24】 对文件上传进行限制,用户只能上传.gif 或.jpeg 文件,文件大小必须小于 20KB。

```
<?php
//接收从表单中 POST 过来的文件数据并处理
if ((( $_FILES["file"]["type"] == "image/gif")
|| ( $_FILES["file"]["type"] == "image/jpeg")
|| ( $_FILES["file"]["type"] == "image/pjpeg"))
&& ( $_FILES["file"]["size"] < 20000))
{
  if ( $_FILES["file"]["error"] > 0)
  {
    echo "错误: " . $_FILES["file"]["error"] . "<br />";
  }
  else
  {
```

```
      echo "上传: " . $_FILES["file"]["name"] . "<br />";
      echo "类型: " . $_FILES["file"]["type"] . "<br />";
      echo "大小: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
      echo "存储为: " . $_FILES["file"]["tmp_name"];
    }
  }
else
{
  echo "无效文件";
}
?>
```

注意：对于 IE，识别 jpg 文件的类型必须是 pjpeg，对于 Firefox，必须是 jpeg。

4. 保存被上传的文件

上面的例子在服务器的 PHP 临时文件夹中创建了一个被上传文件的临时副本。这个临时的副本文件会在脚本结束时消失。要保存被上传的文件，需要把它复制到另外的位置。

【例 10.25】 把上传文件保存到名为 upload 的新文件夹中。

```
<?php
//接收从表单中 POST 过来的文件数据并处理
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
  if ($_FILES["file"]["error"] > 0)
  {
    echo "错误: " . $_FILES["file"]["error"] . "<br />";
  }
  else
  {
    echo "上传: " . $_FILES["file"]["name"] . "<br />";
    echo "类型: " . $_FILES["file"]["type"] . "<br />";
    echo "大小: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "临时文件: " . $_FILES["file"]["tmp_name"] . "<br />";
    if (file_exists("upload/" . $_FILES["file"]["name"]))
    {
      echo $_FILES["file"]["name"] . "已经存在. ";
    }
else
    {
      //移动文件位置
      move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
      //输出新位置
      echo "存储位置: " . "upload/" . $_FILES["file"]["name"];
     }
  }
}
```

```
else
{
  echo "无效文件";
}
?>
```

上面的脚本检测了是否已存在此文件，如果不存在，则把文件复制到指定的文件夹。

注意：这个例子把文件保存到了名为 upload 的新文件夹中。

小结

本章系统介绍了通过 PHP 语言获取和处理所提交表单的数据的方法，并结合数据验证和文件上传的实例通过注释的方式对一些关键的操作方法进行了解释说明。当然，学习是一个循序渐进的过程，虽然 PHP 有许多直观的函数，可以很简单地使用，但是如果没有真正了解它在什么样的状态下该怎么使用，是无法真正灵活掌握表单的各项功能的。只有进一步深入地进行研究，通过不断的练习和实战才能真正完全掌握这些接口方法。

习题

1. HTML 中设置复选框的标签是什么？
2. PHP 表单中 GET 与 POST 提交方法有什么区别？
3. 通过相同的或不同的过滤器来过滤多个变量的函数是什么？
4. 利用 HTML 制作一个邮箱注册申请的网页，用正则表达式来验证用户提交的数据。

第11章 PHP操作数据库

本章要点：

- 连接和关闭数据库；
- 对数据库的操作；
- 对数据表的操作。

关系型数据库是目前信息管理系统的基础软件，在分类存储信息时，关系型数据库非常有用。在市场上的所有数据库中，MySQL 是最受 PHP 程序员青睐的产品。MySQL 是一个小型关系型数据库管理系统，开发者为瑞典 MySQL AB 公司，后被 Sun 公司收购。而 2009 年，Sun 又被 Oracle 收购。MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成本低，以及开放源码等特点，许多采用 PHP 开发的网站都配套选择了 MySQL 作为存储网站数据的数据库。PHP 将对 MySQL 数据库的操作，如连接、断开、查询、修改、删除等，都封装成了函数，以方便开发者操作数据库中的这些数据，使得使用 PHP 开发数据库应用的过程变得极其方便简单。

表 11.1 展现了一个名为 Persons 数据表的例子。

表 11.1　Persons 数据表

LastName	FirstName	Address	Age	City
三	张	中关村	10	北京
四	李	外滩	23	上海
五	王	火车站	20	广州

该表含有三个记录（每个记录是一个人）和 5 个列（LastName，FirstName，Address，Age 以及 City）。

11.1　连接和关闭数据库

操作数据库时，首先必须要与数据库建立连接。PHP 提供了操作 MySQL 的函数库。

11.1.1　连接数据库

在 PHP 中，必须要先通过 mysql_connect()函数完成与数据库的连接，才能访问并处理数据库中的数据。

语法格式如下：

```
mysql_connect(servername,username,password);
```

参数描述：

（1）servername：可选。规定要连接的服务器。默认是 localhost:3306。

（2）username：可选。规定登录所使用的用户名。默认值是拥有服务器进程的用户的名称。

（3）password：可选。规定登录所用的密码。默认是“”。

注释：还存在其他的参数，上面仅列出了最重要的参数。请阅读 PHP 的 MySQL 参考手册，获得更多的细节信息。

【例 11.1】　连接数据库。

在下面的例子中，在一个变量中（$con）存放了在脚本中供稍后使用的连接。如果连接失败，将执行 die 部分：

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (!$con)
{
    die('无法连接到服务器：'. mysql_error());
}
?>
```

如果无法连接 MySQL 数据库，那么就无法继续完成预期的工作。因此，一定要注意监视连接错误并相应地做出反应。PHP 提供了很多内置函数可以用来捕获数据库操作时的错误信息，例如 mysqli_connect_errno()和 mysqli_connect_error()方法。

（1）mysqli_connect_errno()函数返回连接数据库返回的错误号。

（2）mysqli_connect_error()函数返回连接数据库返回的错误代码。

【例 11.2】　连接错误返回值。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (mysqli_connect_errno()) {
    echo '数据库连接错误，错误信息：'.mysqli_connect_error();
    exit();
}
//errno 属性返回数据库操作时的错误号。
//error 属性返回数据库操作时的错误代码。
if ($_mysqli->errno) {
    echo '数据库操作时发生错误，错误代码是：'.$_mysqli->error;
}
?>
```

11.1.2　关闭数据库

PHP 脚本执行结束后，可以使用 mysql_close()函数关闭连接，释放连接资源。

语法格式如下：

```
bool mysql_close ( [resource link_identifier] )
```

如果成功则返回 TRUE,失败则返回 FALSE。

mysql_close()关闭指定的连接标识所关联的到 MySQL 服务器的连接。如果没有指定 link_identifier,则关闭上一个打开的连接。

通常不需要使用 mysql_close(),因为由 mysql_connect 打开的连接会在脚本执行完毕后自动关闭。但若在脚本中间用完后,提倡使用此函数及时连接资源,以提高效率。

【例 11.3】 关闭数据库。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (!$con)
  {
  die('无法连接数据库: ' . mysql_error());
  }
// 管理数据库连接
mysql_close($con);
?>
```

11.2 对数据库的操作

11.2.1 创建数据库

在 MySQL 中创建数据库用 CREATE DATABASE 语句。

语法格式如下:

```
CREATE DATABASE database_name
```

为了让 PHP 执行上面的语句,必须使用 mysql_query()函数。此函数用于向 MySQL 连接发送查询或命令。

【例 11.4】 创建一个名为 my_db 的数据库。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (!$con)
{
  die('无法连接: ' . mysql_error());
}
//执行创建数据库操作
if (mysql_query("CREATE DATABASE my_db",$con))
{
  echo "数据库创建成功";
}
else
{
```

```
  echo "创建数据库失败: " . mysql_error();
}
//关闭数据库连接
mysql_close( $ con);
?>
```

11.2.2 选择数据库

在创建数据库或者对数据库进行其他操作之前,必须首先选择数据库。通过 mysql_select_db()函数选取数据库,使其成为当前数据库。一个数据库成为当前数据库,那么当前所有的操作都是针对它的。

语法格式如下:

```
bool mysql_select_db ( string database_name [, resource link_identifier])
```

如果成功则返回 TRUE,失败则返回 FALSE。

mysql_select_db()设定与指定的连接标识符所关联的服务器上的当前数据库。如果没有指定连接标识符,则使用上一个打开的连接。如果没有打开的连接,本函数将无参数调用 mysql_connect()来尝试打开一个并使用它。

【例 11.5】 选择数据库。

```
<?
 $ lnk = mysql_connect('localhost', 'root', 'adc123')
      or die("无法连接数据库服务器");

// 将数据库 abc 设置为当前数据库
mysql_select_db('abc', $ server_link) or die ('无法选择数据库 abc: ' .
mysql_error());

?>
```

11.2.3 删除数据库

对于已经废弃的数据库,可以将其删除,以节省服务器空间。

语法格式如下:

```
DROP DATABASE 数据库名
```

11.3 对数据表的操作

11.3.1 建立数据表

CREATE TABLE 用于在数据库中创建数据库表。

语法格式如下:

```
CREATE TABLE table_name
```

```
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
...
)
```

为了执行此命令,必须向 mysql_query()函数中添加 CREATE TABLE 语句。

【例 11.6】 创建一个名为 Persons 的表,此表有三列,列名是 FirstName,LastName 以及 Age。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (!$con)
{
  die('无法连接数据库: ' . mysql_error());
}
// 执行创建数据库操作
if (mysql_query("CREATE DATABASE my_db", $con))
{
  echo "数据库创建成功";
}
else
{
  echo "创建数据库失败: " . mysql_error();
}
//选择数据库 my_db
mysql_select_db("my_db", $con);
// 创建数据表 Persons
$sql = "CREATE TABLE Persons
(
    FirstName varchar(15),
    LastName varchar(15),
    Age int
)";
mysql_query($sql, $con);
//关闭数据库连接
mysql_close($con);
?>
```

注意:在创建表之前,必须首先通过 mysql_select_db()函数选取数据库。

11.3.2 向数据库表插入数据

INSERT INTO 语句用于向数据库表添加新记录。

语法格式如下:

```
INSERT INTO table_name VALUES (value1, value2, …)
```

还可以指定希望在其中插入数据的列:

```
INSERT INTO table_name (column1, column2, … ) VALUES (value1, value2, … )
```

注意：SQL 语句对大小写不敏感。INSERT INTO 与 insert into 相同。

在例 11.6 中，创建了一个名为 Persons 的表，有三个列：Firstname，Lastname 以及 Age，下面将在例 11.7 中使用同样的表。

【例 11.7】 在已创建的 Persons 表中添加两条新记录。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (!$con)
{
  die('无法连接数据库：'. mysql_error());
}
//选择数据库 my_db
mysql_select_db("my_db", $con);
//向 Persons 表插入数据
mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('张', '三', '35')");
//向 Persons 表插入数据
mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('李', '四', '33')");
//关闭数据库连接
mysql_close($con);
?>
```

11.3.3 更新数据库表中的数据

UPDATE 语句用于在数据库表中修改数据。

语法格式如下：

```
UPDATE table_name SET column_name = new_value WHERE column_name = some_value
```

【例 11.8】 在已建立的 Persons 表中更新数据。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (!$con)
{
  die('无法连接数据库：'. mysql_error());
}
//选择数据库 my_db
mysql_select_db("my_db", $con);
//更新数据库中的数据
mysql_query("UPDATE Persons SET Age = '36'
WHERE FirstName = '张' AND LastName = '三'");
//关闭数据库连接
mysql_close($con);
?>
```

更新后 Persons 数据表中数据如表 11.2 所示。

表 11.2 更新后的 Persons 数据表

FirstName	LastName	Age
张	三	36
李	四	33

11.3.4 删除数据库中的数据

DELETE FROM 语句用于从数据库表中删除记录。

语法格式如下：

```
DELETE FROM table_name WHERE column_name = some_value
```

【例 11.9】 删除 Persons 表中所有 LastName='Griffin'的记录。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost","root","abc123");
if (!$con)
{
  die('无法连接数据库：'. mysql_error());
}
//选择数据库 my_db
mysql_select_db("my_db", $con);
//删除数据库中的数据
mysql_query("DELETE FROM Persons WHERE FirstName = '张'");
//关闭数据库连接
mysql_close($con);
?>
```

执行删除程序后的 Persons 数据表中数据如表 11.3 所示。

表 11.3 执行删除后的 Persons 表

FirstName	LastName	Age
李	四	33

11.3.5 从数据库表中选取数据

SELECT 语句用于从数据库中选取数据。

语法格式如下：

```
SELECT column_name(s) FROM table_name
```

【例 11.10】 选取存储在 Persons 表中的所有数据。

```
<?php
//传入连接 MySQL 的主机、用户、密码、数据库
```

```
$con = mysql_connect("localhost","root","abc123");
if (!$con)
{
  die('无法连接数据库: ' . mysql_error());
}
//选择数据库 my_db
mysql_select_db("my_db", $con);
//执行查询
$result = mysql_query("SELECT * FROM Persons");
//从查询结果中循环输出
while($row = mysql_fetch_array($result))
{
  echo $row['FirstName'] . "" . $row['LastName'];
  echo "<br />";
}
//关闭数据库连接
mysql_close($con);
?>
```

在例 11.10 中，$result 变量中存放着由 mysql_query()函数返回的数据，使用 mysql_fetch_array()函数以数组的形式从记录集返回第一行。随后对 mysql_fetch_array()函数的调用都会返回记录集中的下一行。while loop 语句会循环记录集中的所有记录。为了输出每行的值，使用了 PHP 的 $row 变量($row['FirstName']和 $row['LastName'])。

以上代码将输出以下内容：

```
张三
李四
```

11.3.6 获取记录个数

可以通过调用 mysql_num_rows()查看 SELECT 语句的查询结果的行数。

语法格式如下：

```
mysql_num_rows()
```

【例 11.11】 获取记录个数。

```
<?
//传入连接 MySQL 的主机、用户、密码、数据库
$con = mysql_connect("localhost", "root", "abc123");
//选择数据库
mysql_select_db("db1", $con);
//查询记录
$result = mysql_query("SELECT * FROM table1", $con);
//获取记录数量
$num_rows = mysql_num_rows($result);
//输出记录数量
echo "$num_rows 行\n";
?>
```

mysql_num_rows 命令仅对 SELECT 语句有效。

要取得被 INSERT、UPDATE 或者 DELETE 查询所影响到的行的数目，需要使用 mysql_affected_rows()函数。

语法格式如下：

```
mysql_affected_rows() (resource result)
```

【例 11.12】 获取受影响记录个数。

```
<?
//传入连接 MySQL 的主机、用户、密码、数据库
$con= mysql_connect("localhost", "root", "abc123");
//选择数据库
mysql_select_db("db1", $con);
//插入数据
$result = mysql_query("INSERT INTO table1 (col1,col2) values ('a','b')", $con);
//获取受影响的行数
$num_rows = mysql_affected_rows($result);
//输出受影响的行数
echo "$num_rows 行\n";
?>
```

小结

本章系统介绍了通过 PHP 语言连接、选择、读取数据库表数据的常用操作，并结合实例对一些关键接口方法进行了说明。本章的内容主要结合 MySQL 数据库进行相关介绍，在实际场景中，读者可能会遇到需要操作其他类型数据库的情况，但具体语法基本类似，仍然可以参考本章学习的内容。

习题

1. 写出启动 MySQL 和关闭 MySQL 的命令。
2. 建立 student 数据库，建立 class1、class2 数据表，在 class1 中建立 num、name、sex 字段，并写入两条记录。

第12章 PHP动态网站设计实例

本章要点：

- 概要设计；
- 代码实现。

在以上的章节中，对于动态网站的开发所涉及的内容进行了详述。在本章中将综合所学内容，制作一个网上商城管理系统的实例。

12.1 概要设计

12.1.1 系统功能及逻辑结构

网上商城系统的业务逻辑比较简单，可以参考互联网上现成的站点，如天猫、京东等，其主体功能包括前台购物结算界面和后台管理两大部分。

前台部分由用户使用，主要包括用户管理（用户登录、注册），商品浏览，购物车管理，个人账户管理等。

后台部分由管理员使用，主要包括管理员身份验证，商品管理，订单管理，会员管理，系统管理等。

整个网上商城的逻辑结构图如图 12.1 所示。

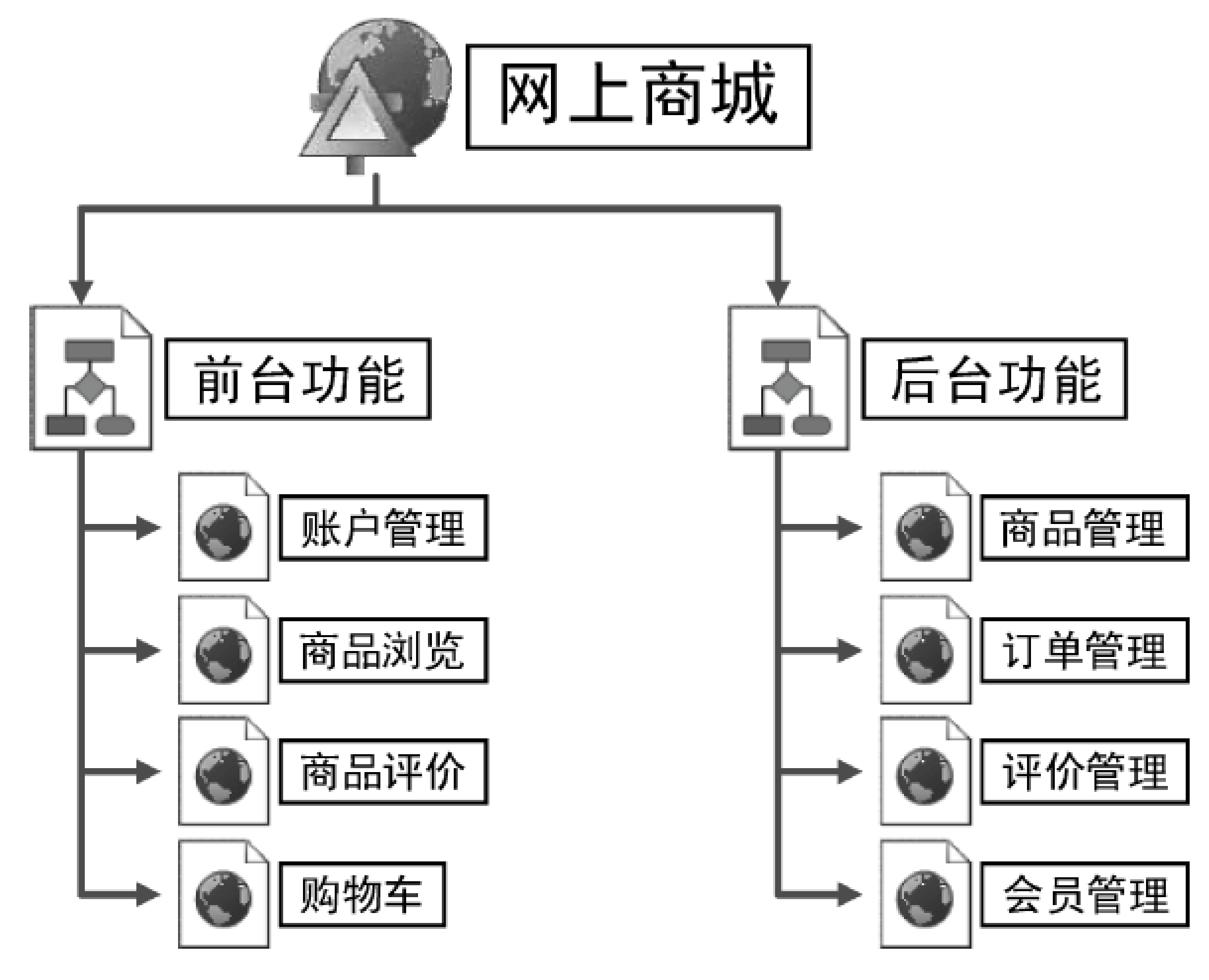


图 12.1　在线商城逻辑结构图

12.1.2 数据库设计

网上商城的数据库设计中,包括 20 个表的建立。表 12.1~表 12.20 详细介绍了各个表的字段名称,代码以及数据类型。

商品信息表如表 12.1 所示。

表 12.1 商品信息表(products)

字段名称	代码	数据类型	是否为空
商品编号(自增)	products_id	int(5)	DEFAULT'0'NOTNULL
商品名称	products_name	varchar(32)	DEFAULT''NOTNULL
商品描述	products_description	text	
商品库存数量	products_quantity	int(4)	DEFAULT'0'NOTNULL
商品模型	products_model	varchar(12)	
商品图片	products_image	varchar(64)	
商品地址	products_url	varchar(255)	
商品报价	products_price	decimal(8,2)	DEFAULT'0.00'NOTNULL
商品上架日期	products_date_added	varchar(8)	
商品查看次数	products_viewed	int(5)	DEFAULT'0'

商品订单信息表如表 12.2 所示。

表 12.2 商品订单信息表(orders)

字段名称	代码	数据类型	是否为空
订单编号(自增)	orders_id	int(5)	DEFAULT'0'NOTNULL
顾客编号	customers_id	int(5)	DEFAULT'0'NOTNULL
顾客姓名	customers_name	varchar(64)	DEFAULT''NOTNULL
顾客地址	customers_street_address	varchar(64)	DEFAULT''NOTNULL
顾客城市	customers_city	varchar(32)	DEFAULT''NOTNULL
顾客邮编	customers_postcode	varchar(8)	DEFAULT''NOTNULL
顾客省份	customers_state	varchar(32)	
顾客国家	customers_country	varchar(32)	DEFAULT''NOTNULL
顾客电话	customers_telephone	varchar(32)	DEFAULT''NOTNULL
顾客邮件地址	customers_email_address	varchar(96)	DEFAULT''NOTNULL
收货人姓名	delivery_name	varchar(64)	DEFAULT''NOTNULL
收货地址	delivery_street_address	varchar(64)	DEFAULT''NOTNULL
收货城市	delivery_city	varchar(32)	DEFAULT''NOTNULL
收货邮编	delivery_postcode	varchar(8)	DEFAULT''NOTNULL
收货省份	delivery_state	varchar(32)	
收货国家	delivery_country	varchar(32)	DEFAULT''NOTNULL
支付方式	payment_method	varchar(12)	DEFAULT''NOTNULL
类型	cc_type	varchar(20)	
所有者	cc_owner	varchar(64)	
编号	cc_number	varchar(32)	
过期	cc_expires	varchar(4)	
购买日期	date_purchased	varchar(8)	
商品缴税	products_tax	int(2)	

用户信息表如表12.3所示。

表12.3　用户信息表(olm_cart)

字段名称	代　码	数据类型	是否为空
客户编号(自增)	customers_id	int(5)	DEFAULT'0'NOTNULL
客户性别	customers_gender	char(1)	DEFAULT''NOTNULL
客户姓	customers_firstname	varchar(32)	DEFAULT''NOTNULL
客户名	customers_lastname	varchar(32)	DEFAULT''NOTNULL
客户生日	customers_dob	varchar(8)	DEFAULT''NOTNULL
客户 E-mail 地址	customers_email_address	varchar(96)	DEFAULT''NOTNULL
客户实际地址	customers_street_address	varchar(64)	DEFAULT''NOTNULL
客户邮编	customers_postcode	varchar(8)	DEFAULT''NOTNULL
客户城市	customers_city	varchar(32)	DEFAULT''NOTNULL
客户省份	customers_state	varchar(32)	
客户国家	customers_country	varchar(32)	DEFAULT''NOTNULL
客户电话	customers_telephone	varchar(32)	DEFAULT''NOTNULL
客户传真	customers_fax	varchar(32)	
客户密码	customers_password	varchar(12)	DEFAULT''NOTNULL

供应商信息表如表12.4所示。

表12.4　供应商信息表(manufacturers)

字段名称	代　码	数据类型	是否为空
供应商编号	manufacturers_id	int(5)	DEFAULT'0'NOTNULL
供应商名称	manufacturers_name	varchar(32)	DEFAULT''NOTNULL
供应商地址	manufacturers_location	tinyint(1)	DEFAULT'0'NOTNULL
供应商 LOGO	manufacturers_image	varchar(64)	

商品特价信息表如表12.5所示。

表12.5　商品特价信息表(specials)

字段名称	代　码	数据类型	是否为空
特价编号	specials_id	int(5)	DEFAULT'0'NOTNULL
商品编号	products_id	int(5)	DEFAULT'0'NOTNULL
优惠价格	specials_new_products_price	decimal(8,2)	DEFAULT'0.00'NOTNULL
优惠日期	specials_date_added	char(8)	

商品评价信息表如表12.6所示。

表12.6　商品评价信息表(reviews_extra)

字段名称	代　码	数据类型	是否为空
评价编号	reviews_id	int(5)	DEFAULT'0'NOTNULL
商品编号	products_id	int(5)	DEFAULT'0'NOTNULL
客户编号	customers_id	int(5)	DEFAULT'0'NOTNULL
评论时间	date_added	char(8)	DEFAULT''NOTNULL
评价阅读次数	reviews_read	int(5)	

商品评价详细信息表如表12.7所示。

表12.7 商品评价详细信息表(reviews)

字段名称	代码	数据类型	是否为空
评价编号	reviews_id	int(5)	DEFAULT'0'NOTNULL
详细评价	reviews_text	text	NOTNULL
评分	reviews_rating	int(1)	

顾客详细信息表如表12.8所示。

表12.8 顾客详细信息表(customs_info)

字段名称	代码	数据类型	是否为空
客户信息编号	customers_info_id	int(5)	DEFAULT'0'NOTNULL
客户最后登录时间	customers_info_date_of_last_logon	char(8)	
客户登录次数	customers_info_number_of_logons	int(5)	
客户注册时间	customers_info_date_account_created	char(8)	
客户信息修改时间	customers_info_date_account_last_modified	char(8)	

顾客购物篮信息表如表12.9所示。

表12.9 顾客购物篮信息表(customs_basket)

字段名称	代码	数据类型	是否为空
订单条目编号	customers_basket_id	int(5)	DEFAULT'0'NOTNULL
客户编号	customers_id	int(5)	DEFAULT'0'NOTNULL
商品编号	products_id	int(5)	DEFAULT'0'NOTNULL
商品数量	customers_basket_quantity	int(2)	DEFAULT'0'NOTNULL
商品购买日期	customers_basket_date_added	char(8)	

送货地址信息表如表12.10所示。

表12.10 送货地址信息表(address_book)

字段名称	代码	数据类型	是否为空
送货地址编号	address_book_id	int(5)	DEFAULT'0'NOTNULL
收货人性别	entry_gender	char(1)	DEFAULT''NOTNULL
收货人姓	entry_firstname	varchar(32)	DEFAULT''NOTNULL
收货人名	entry_lastname	varchar(32)	DEFAULT''NOTNULL
收货人地址	entry_street_address	varchar(64)	DEFAULT''NOTNULL
收货人邮编	entry_postcode	varchar(8)	DEFAULT''NOTNULL
收货人城市	entry_city	varchar(32)	DEFAULT''NOTNULL
收货人省份	entry_state	varchar(32)	
收货人国家	entry_country	varchar(32)	DEFAULT''NOTNULL

顾客送货地址关系表如表12.11所示。

表12.11 顾客送货地址关系表(address_book_to_customers)

字段名称	代码	数据类型	是否为空
关系编号	address_book_to_customers_id	int(5)	DEFAULT'0'NOTNULL
送货地址编号	address_book_id	int(5)	DEFAULT'0'NOTNULL
客户编号	customers_id	int(5)	DEFAULT'0'NOTNULL

订单明细信息表如表 12.12 所示。

表 12.12　订单明细信息表(orders_product)

字段名称	代　码	数据类型	是 否 为 空
明细编号	orders_products_id	int(5)	DEFAULT'0'NOTNULL
订单编号	orders_id	int(5)	DEFAULT'0'NOTNULL
商品编号	products_id	int(5)	DEFAULT'0'NOTNULL
商品名称	products_name	varchar(64)	DEFAULT''NOTNULL
商品价格	products_price	decimal(8,2)	DEFAULT'0.00'NOTNULL
商品数量	products_quantity	int(2)	DEFAULT'0'NOTNULL

商品供应商关系表如表 12.13 所示。

表 12.13　商品供应商关系表(products_to_manufacturers)

字段名称	代　码	数据类型	是 否 为 空
关系编号	products_to_manufacturers_id	int(5)	DEFAULT'0'NOTNULL
商品编号	products_id	int(5)	DEFAULT'0'NOTNULL
供应商编号	manufacturers_id	int(5)	DEFAULT'0'NOTNULL

供应商分类关系表如表 12.14 所示。

表 12.14　供应商分类关系表(manufacturers_to_category)

字段名称	代　码	数据类型	是 否 为 空
关系编号	manufacturers_to_category_id	int(5)	DEFAULT'0'NOTNULL
供应商编号	manufacturers_id	int(5)	DEFAULT'0'NOTNULL
根分类编号	category_top_id	int(5)	DEFAULT'0'NOTNULL

商品顶级分类信息表如表 12.15 所示。

表 12.15　商品顶级分类信息表(category_top)

字段名称	代　码	数据类型	是 否 为 空
根分类编号	category_top_id	int(5)	DEFAULT'0'NOTNULL
根分类名称	category_top_name	varchar(32)	DEFAULT''NOTNULL
排序	sort_order	int(3)	
根分类图片	category_image	varchar(64)	

商品分类索引表如表 12.16 所示。

表 12.16　商品分类索引表(category_index)

字 段 名 称	代　码	数据类型	是 否 为 空
分类索引编号	category_index_id	int(5)	DEFAULT'0'NOTNULL
分类索引名称	category_index_name	varchar(32)	DEFAULT''NOTNULL
查询 SQL	sql_select	varchar(16)	DEFAULT''NOTNULL

商品顶级分类与索引关系表如表12.17所示。

表12.17 商品顶级分类与索引关系表(category_index_to_top)

字段名称	代　码	数据类型	是否为空
关系编号	category_index_to_top_id	int(5)	DEFAULT'0'NOTNULL
根分类编号	category_top_id	int(5)	DEFAULT'0'NOTNULL
分类索引编号	category_index_id	int(5)	
排序	sort_order	int(3)	

商品子分类信息表如表12.18所示。

表12.18 商品子分类信息表(subcategories)

字段名称	代　码	数据类型	是否为空
子分类编号	subcategories_id	int(5)	DEFAULT'0'NOTNULL
子分类名称	subcategories_name	varchar(32)	DEFAULT''NOTNULL
子分类图片	subcategories_image	varchar(64)	

分类父子关系表如表12.19所示。

表12.19 分类父子关系表(subcategories_to_category)

字段名称	代　码	数据类型	是否为空
关系编号	subcategories_to_category_id	int(5)	DEFAULT'0'NOTNULL
子分类编号	subcategories_id	int(5)	DEFAULT'0'NOTNULL
父分类编号	category_top_id	int(5)	DEFAULT'0'NOTNULL

商品分类关系表如表12.20所示。

表12.20 商品分类关系表(products_to_subcategories)

字段名称	代　码	数据类型	是否为空
关系编号	products_to_subcategories_id	int(5)	DEFAULT'0'NOTNULL
商品编号	products_id	int(5)	DEFAULT'0'NOTNULL
分类编号	subcategories_id	int(5)	DEFAULT'0'NOTNULL

12.1.3 系统运行环境要求

硬件环境要求如下。

CPU：1 GHz以上。

内存：512MB以上。

分辨率：支持VGA或分辨率更高的监视器。

色彩：256色以上。

操作系统：Microsoft Windows。

软件环境要求如下。

操作系统：Windows XP系列。

数据库：MySQL 5.0。

服务器：Apache。

开发语言：PHP。

12.2 代码实现

最终实现的网上商城系统共包含近一千个 PHP 代码文件，以及相应的样式表和 JS 脚本、图片等。限于篇幅所限，本书仅选取了其中几个核心模块的代码供读者参考。网上商城系统的运行效果如图 12.2 所示。

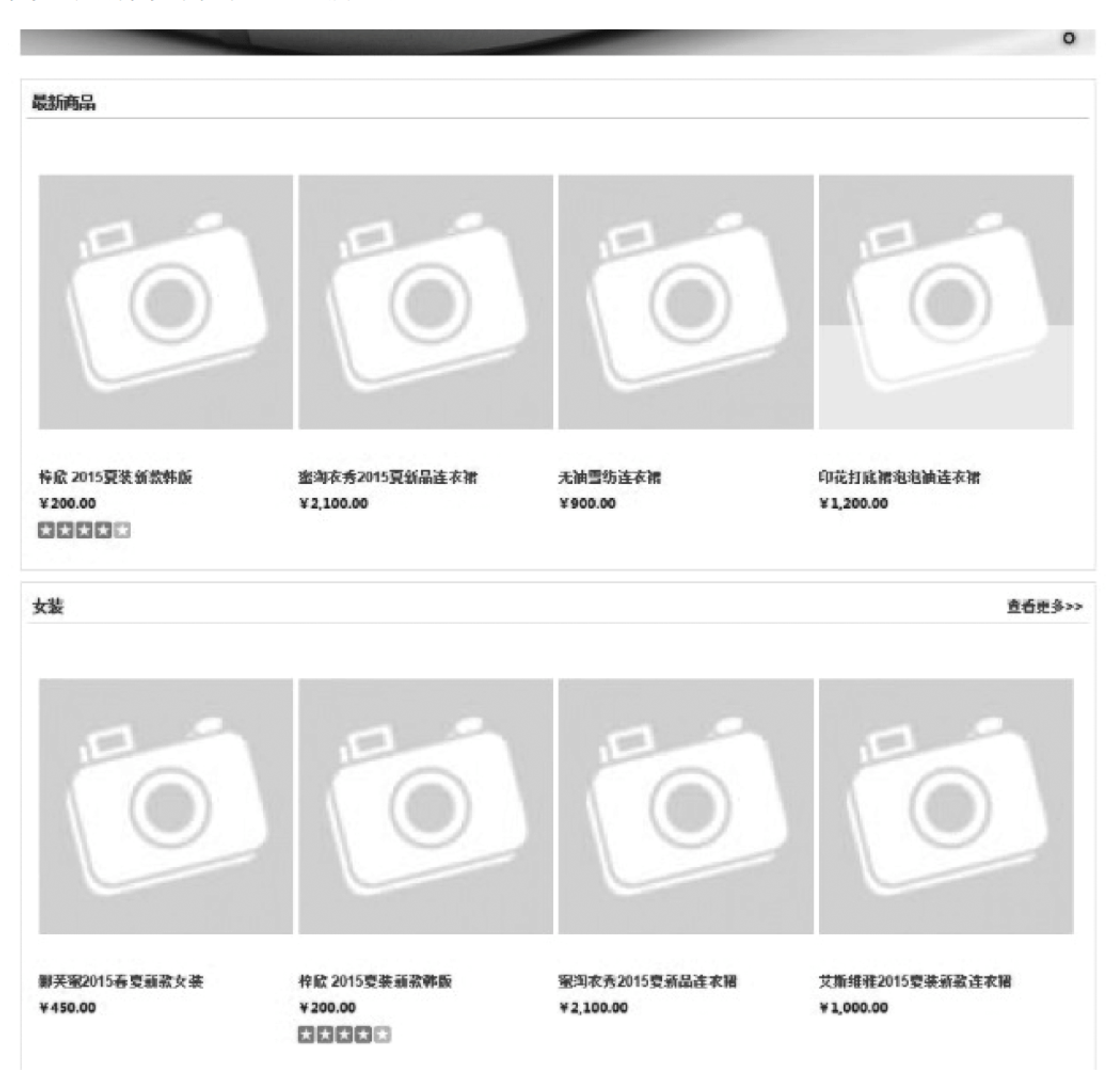


图 12.2 网上商城最终运行效果

12.2.1 注册账号

注册账号功能由 register. php、account. php 和 success. php 这三个核心 PHP 页面组成，其中，register. php 用于填写注册信息，account. php 用于接收和处理注册信息，success. php 则用于注册成功后返回注册信息。

【例 12.1】 register.php 的核心源代码。

```
<?php
//定义用户控制类
class ControllerAffiliateRegister extends Controller {
    private $error = array();

    public function index() {
        if ($this->affiliate->isLogged()) {
            $this->redirect($this->url->link('affiliate/account', '', 'SSL'));
    }

    $this->load_language('affiliate/register');

    $this->document->setTitle($this->language->get('heading_title'));

    $this->load->model('affiliate/affiliate');

    if (($this->request->server['REQUEST_METHOD'] == 'POST') && $this->validate()) {
    $this->model_affiliate_affiliate->addAffiliate($this->request->post);

    $this->affiliate->login($this->request->post['email'], $this->request->post
['password']);

    $this->redirect($this->url->link('affiliate/success'));
        }

    $this->data['breadcrumbs'] = array();

    $this->data['breadcrumbs'][] = array(
    'text'       => $this->language->get('text_home'),
            'href'       => $this->url->link('common/home'),
    'separator' => false
    );

    $this->data['breadcrumbs'][] = array(
    'text'       => $this->language->get('text_account'),
            'href'       => $this->url->link('affiliate/account', '', 'SSL'),
    'separator' => $this->language->get('text_separator')
    );

    $this->data['breadcrumbs'][] = array(
    'text'       => $this->language->get('text_register'),
            'href'       => $this->url->link('affiliate/register', '', 'SSL'),
    'separator' => $this->language->get('text_separator')
    );

    $this->data['text_account_already'] = sprintf($this->language->get('text_account_
already'), $this->url->link('affiliate/login', '', 'SSL'));

    $errors = array(
```

```
    'warning' => 'error_warning',
'firstname' => 'error_firstname',
'email' => 'error_email',
'telephone' => 'error_telephone',
'password' => 'error_password',
'confirm' => 'error_confirm',
'address_1' => 'error_address_1',
'city' => 'error_city',
'postcode' => 'error_postcode',
'country' => 'error_country',
'zone' => 'error_zone',

);

foreach ($errors as $key => $value) {
    if (isset($this->error[$key])) {
        $this->data[$value] = $this->error[$key];
    } else {
        $this->data[$value] = '';
    }
}

    $this->data['action'] = $this->url->link('affiliate/register', '', 'SSL');

    if (isset($this->request->post['firstname'])) {
    $this->data['firstname'] = $this->request->post['firstname'];
    } else {
        $this->data['firstname'] = '';
    }

    if (isset($this->request->post['email'])) {
    $this->data['email'] = $this->request->post['email'];
    } else {
        $this->data['email'] = '';
    }

    if (isset($this->request->post['telephone'])) {
    $this->data['telephone'] = $this->request->post['telephone'];
    } else {
        $this->data['telephone'] = '';
    }

    if (isset($this->request->post['fax'])) {
    $this->data['fax'] = $this->request->post['fax'];
    } else {
        $this->data['fax'] = '';
    }

    if (isset($this->request->post['company'])) {
    $this->data['company'] = $this->request->post['company'];
```

```
    } else {
        $this->data['company'] = '';
    }

    if (isset($this->request->post['website'])) {
    $this->data['website'] = $this->request->post['website'];
    } else {
        $this->data['website'] = '';
    }

    if (isset($this->request->post['address_1'])) {
    $this->data['address_1'] = $this->request->post['address_1'];
    } else {
        $this->data['address_1'] = '';
    }

    if (isset($this->request->post['address_2'])) {
    $this->data['address_2'] = $this->request->post['address_2'];
    } else {
        $this->data['address_2'] = '';
    }

    if (isset($this->request->post['postcode'])) {
    $this->data['postcode'] = $this->request->post['postcode'];
    } else {
        $this->data['postcode'] = '';
    }

    if (isset($this->request->post['city_id'])) {
    $this->data['city_id'] = $this->request->post['city_id'];
    } else {
        $this->data['city_id'] = '';
    }


$this->data['country_id'] = $this->config->get('config_country_id');


if (isset($this->request->post['zone_id'])) {
    $this->data['zone_id'] = $this->request->post['zone_id'];
    } else {
    $this->data['zone_id'] = '';
}

    $this->load->model('localisation/country');

$this->data['countries'] = $this->model_localisation_country->getCountries();

    if (isset($this->request->post['tax'])) {
    $this->data['tax'] = $this->request->post['tax'];
    } else {
        $this->data['tax'] = '';
    }
```

```
if (isset($this->request->post['payment'])) {
$this->data['payment'] = $this->request->post['payment'];
} else {
    $this->data['payment'] = 'cheque';
}

if (isset($this->request->post['cheque'])) {
$this->data['cheque'] = $this->request->post['cheque'];
} else {
    $this->data['cheque'] = '';
}

if (isset($this->request->post['paypal'])) {
$this->data['paypal'] = $this->request->post['paypal'];
} else {
    $this->data['paypal'] = '';
}

if (isset($this->request->post['bank_name'])) {
$this->data['bank_name'] = $this->request->post['bank_name'];
} else {
    $this->data['bank_name'] = '';
}

if (isset($this->request->post['bank_branch_number'])) {
$this->data['bank_branch_number'] = $this->request->post['bank_branch_number'];
} else {
    $this->data['bank_branch_number'] = '';
}

if (isset($this->request->post['bank_swift_code'])) {
$this->data['bank_swift_code'] = $this->request->post['bank_swift_code'];
} else {
    $this->data['bank_swift_code'] = '';
}

if (isset($this->request->post['bank_account_name'])) {
$this->data['bank_account_name'] = $this->request->post['bank_account_name'];
} else {
    $this->data['bank_account_name'] = '';
}

if (isset($this->request->post['bank_account_number'])) {
$this->data['bank_account_number'] = $this->request->post['bank_account_
number'];
} else {
    $this->data['bank_account_number'] = '';
}

if (isset($this->request->post['password'])) {
$this->data['password'] = $this->request->post['password'];
} else {
    $this->data['password'] = '';
}
```

```
        if (isset($this->request->post['confirm'])) {
        $this->data['confirm'] = $this->request->post['confirm'];
        } else {
            $this->data['confirm'] = '';
        }

        if ($this->config->get('config_affiliate_id')) {
            $this->load->model('catalog/information');

             $information_info = $this->model_catalog_information->getInformation
($this->config->get('config_affiliate_id'));

            if ($information_info) {
                $this->data['text_agree'] = sprintf($this->language->get('text_agree
'), $this->url->link('information/information/info', 'information_id=' . $this->config
->get('config_affiliate_id'), 'SSL'), $information_info['title'], $information_info['title
']);
            } else {
                $this->data['text_agree'] = '';
            }
        } else {
            $this->data['text_agree'] = '';
        }

        if (isset($this->request->post['agree'])) {
        $this->data['agree'] = $this->request->post['agree'];
        } else {
            $this->data['agree'] = false;
        }

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
affiliate/register.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
affiliate/register.tpl';
        } else {
            $this->template = 'default/template/affiliate/register.tpl';
        }

        $this->children = array(
            'common/column_left',
            'common/column_right',
            'common/content_top',
            'common/content_bottom',
            'common/footer',
            'common/header'
        );

        $this->response->setOutput($this->render());
    }

    private function validate() {
    if ((strlen(utf8_decode($this->request->post['firstname'])) < 1) || (strlen(utf8_
decode($this->request->post['firstname'])) > 32)) {
```

```
        $this->error['firstname'] = $this->language->get('error_firstname');
    }

    if ((strlen(utf8_decode($this->request->post['email'])) > 96) || !preg_match('/^[^\
@]+@.*\.[a-z]{2,6}$/i', $this->request->post['email'])) {
        $this->error['email'] = $this->language->get('error_email');
    }

    if ($this->model_affiliate_affiliate->getTotalAffiliatesByEmail($this->request->
post['email'])) {
        $this->error['warning'] = $this->language->get('error_exists');
    }

    if ((strlen(utf8_decode($this->request->post['telephone'])) < 1) || (strlen(utf8_
decode($this->request->post['telephone'])) > 32)) {
        $this->error['telephone'] = $this->language->get('error_telephone');
    }

    if ((strlen(utf8_decode($this->request->post['address_1'])) < 1) || (strlen(utf8_
decode($this->request->post['address_1'])) > 128)) {
        $this->error['address_1'] = $this->language->get('error_address_1');
    }

    if ($this->request->post['city_id'] == '') {
        $this->error['city'] = $this->language->get('error_city');
    }

        $this->load->model('localisation/country');

        $country_info = $this->model_localisation_country->getCountry($this->
config->get('config_country_id'));

        if ($country_info && $country_info['postcode_required'] && (strlen(utf8_decode
($this->request->post['postcode'])) < 2) || (strlen(utf8_decode($this->request->post
['postcode'])) > 10)) {
            $this->error['postcode'] = $this->language->get('error_postcode');
        }

    if ($this->request->post['zone_id'] == '') {
        $this->error['zone'] = $this->language->get('error_zone');
    }

    if ((strlen(utf8_decode($this->request->post['password'])) < 4) || (strlen(utf8_
decode($this->request->post['password'])) > 20)) {
        $this->error['password'] = $this->language->get('error_password');
    }

    if ($this->request->post['confirm'] != $this->request->post['password']) {
        $this->error['confirm'] = $this->language->get('error_confirm');
    }
```

```
        if ($this->config->get('config_affiliate_id')) {
            $this->load->model('catalog/information');

            $information_info = $this->model_catalog_information->getInformation
($this->config->get('config_affiliate_id'));

            if ($information_info && !isset($this->request->post['agree'])) {
            $this->error['warning'] = sprintf($this->language->get('error_agree'),
$information_info['title']);
            }
        }

    if (!$this->error) {
        return true;
    } else {
        return false;
    }
    }

}
?>
```

注册页面效果如图 12.3 所示。



图 12.3　注册页面效果

12.2.2　登录注销

为了实现身份辨识,注册用户需要输入用户名和密码登录后才能进行购物操作。此部分的功能主要包括:登录、注销、忘记密码等组成,分别由 login.php、logout.php、forgotten.

php 和 password. php 等页面组成。

【例 12. 2】 login. php 的源代码。

```
<?php
class ControllerAffiliateLogin extends Controller {
    private $error = array();

    public function index() {
        if ($this->affiliate->isLogged()) {
        $this->redirect($this->url->link('affiliate/account', '', 'SSL'));
    }

    $this->language->load('affiliate/login');

    $this->document->setTitle($this->language->get('heading_title'));

        if (($this->request->server['REQUEST_METHOD'] == 'POST') && isset($this->
request->post['email']) && isset($this->request->post['password']) && $this->validate
()) {
            if (isset($this->request->post['redirect'])) {
                $this->redirect(str_replace('&amp;', '&', $this->request->post
['redirect']));
            } else {
                $this->redirect($this->url->link('affiliate/account', '', 'SSL'));
            }
        }

    $this->data['breadcrumbs'] = array();

    $this->data['breadcrumbs'][] = array(
    'text'       => $this->language->get('text_home'),
            'href'       => $this->url->link('common/home'),
    'separator' => false
    );

    $this->data['breadcrumbs'][] = array(
    'text'       => $this->language->get('text_account'),
            'href'       => $this->url->link('affiliate/account', '', 'SSL'),
    'separator' => $this->language->get('text_separator')
    );

    $this->data['breadcrumbs'][] = array(
    'text'       => $this->language->get('text_login'),
            'href'       => $this->url->link('affiliate/login', '', 'SSL'),
    'separator' => $this->language->get('text_separator')
    );

    $this->data['heading_title'] = $this->language->get('heading_title');

        $this->data['text_description'] = sprintf($this->language->get('text_
```

```
description'), $this->config->get('config_name'), $this->config->get('config_name'),
$this->config->get('config_commission') . '%');
        $this->data['text_new_affiliate'] = $this->language->get('text_new_affiliate');
    $this->data['text_register_account'] = $this->language->get('text_register_
account');
        $this->data['text_returning_affiliate'] = $this->language->get('text_
returning_affiliate');
        $this->data['text_i_am_returning_affiliate'] = $this->language->get('text_i_
am_returning_affiliate');
    $this->data['text_forgotten'] = $this->language->get('text_forgotten');

    $this->data['entry_email'] = $this->language->get('entry_email');
    $this->data['entry_password'] = $this->language->get('entry_password');

    $this->data['button_continue'] = $this->language->get('button_continue');
        $this->data['button_login'] = $this->language->get('button_login');

        if (isset($this->error['warning'])) {
            $this->data['error_warning'] = $this->error['warning'];
        } else {
            $this->data['error_warning'] = '';
        }

        $this->data['action'] = $this->url->link('affiliate/login', '', 'SSL');
        $this->data['register'] = $this->url->link('affiliate/register', '', 'SSL');
        $this->data['forgotten'] = $this->url->link('affiliate/forgotten', '', 'SSL');

        if (isset($this->request->post['redirect'])) {
            $this->data['redirect'] = $this->request->post['redirect'];
        } elseif (isset($this->session->data['redirect'])) {
        $this->data['redirect'] = $this->session->data['redirect'];

            unset($this->session->data['redirect']);
    } else {
            $this->data['redirect'] = '';
        }

        if (isset($this->session->data['success'])) {
        $this->data['success'] = $this->session->data['success'];

            unset($this->session->data['success']);
        } else {
            $this->data['success'] = '';
        }

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
affiliate/login.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
affiliate/login.tpl';
        } else {
            $this->template = 'default/template/affiliate/login.tpl';
```

```
        }

        $this->children = array(
            'common/column_left',
            'common/column_right',
            'common/content_top',
            'common/content_bottom',
            'common/footer',
            'common/header'
        );

        $this->response->setOutput($this->render());
    }

    private function validate() {
    if (!$this->affiliate->login($this->request->post['email'], $this->request->
post['password'])) {
        $this->error['warning'] = $this->language->get('error_login');
    }

    if (!$this->error) {
        return true;
    } else {
        return false;
    }
    }
}
?>
```

用户登录页面效果如图 12.4 所示。



图 12.4　用户登录页面效果

12.2.3 浏览商品

浏览商品功能主要由 category. php、product. php 和 search. php、review. php 等几个页面组成，可以按分类的方式查看对应的商品列表，并从中选择希望购买的商品。用户也可以通过 reviews. php 执行查看和提交商品评价的功能，顾客在购物完成后还可以在商品下添加相应的评语和评分。

下面列出的是 category. php 的源码，从中可以看到如何根据分类与商品的关系获取某分类下的特定商品列表等逻辑的具体实现。

【例 12.3】 category. php 的源码。

```
<?php
class ControllerProductCategory extends Controller {
    public function index() {
        $ this -> load_language('product/category');

        $ this -> load -> model('catalog/category');

        $ this -> load -> model('catalog/product');

        $ this -> load -> model('tool/image');

        if (isset( $ this -> request -> get['sort'])) {
            $ sort  =  $ this -> request -> get['sort'];
        } else {
            $ sort  =  'p. sort_order';
        }

        if (isset( $ this -> request -> get['order'])) {
            $ order  =  $ this -> request -> get['order'];
        } else {
            $ order  =  'ASC';
        }

        if (isset( $ this -> request -> get['page'])) {
            $ page  =  $ this -> request -> get['page'];
        } else {
            $ page  =  1;
        }

        if (isset( $ this -> request -> get['limit'])) {
            $ limit  =  $ this -> request -> get['limit'];
        } else {
            $ limit  =  $ this -> config -> get('config_catalog_limit');
        }

        $ this -> data['breadcrumbs']  =  array();

        $ this -> data['breadcrumbs'][]  =  array(
        'text'        =>  $ this -> language -> get('text_home'),
            'href'        =>  $ this -> url -> link('common/home'),
```

```
        'separator' => false
        );

        if (isset($this->request->get['path'])) {
            $path = '';

            $parts = explode('_', (string)$this->request->get['path']);

            foreach ($parts as $path_id) {
                if (!$path) {
                    $path = $path_id;
                } else {
                    $path .= '_' . $path_id;
                }

                $category_info = $this->model_catalog_category->getCategory($path_id);

                if ($category_info) {
                $this->data['breadcrumbs'][] = array(
                    'text'      => $category_info['name'],
                        'href'      => $this->url->link('product/category', 'path=' .
$path),
                'separator' => $this->language->get('text_separator')
            );
                }
            }

            $category_id = array_pop($parts);
        } else {
            $category_id = 0;
        }

        $category_info = $this->model_catalog_category->getCategory($category_id);

        if ($category_info) {
            if(isset($category_info['meta_title'])&&$category_info['meta_title']!='')
                $this->document->setTitle($category_info['meta_title']);
            else
                $this->document->setTitle($category_info['name']);

            $this->document->setDescription($category_info['meta_description']);
            $this->document->setKeywords($category_info['meta_keyword']);

            $this->data['heading_title'] = $category_info['name'];

            $this->data['text_compare'] = sprintf($this->language->get('text_compare'),
(isset($this->session->data['compare']) ? count($this->session->data['compare']) :
0));

            if ($category_info['image']) {
                $this->data['thumb'] = $this->model_tool_image->resize($category_
info['image'], $this->config->get('config_image_category_width'), $this->config->get
('config_image_category_height'));
            } else {
```

```
            $this->data['thumb'] = '';
        }

        $this->data['description'] = html_entity_decode($category_info['description'],
ENT_QUOTES, 'UTF-8');
        $this->data['compare'] = $this->url->link('product/compare');

        $url = '';

        if (isset($this->request->get['sort'])) {
            $url .= '&sort=' . $this->request->get['sort'];
        }

        if (isset($this->request->get['order'])) {
            $url .= '&order=' . $this->request->get['order'];
        }

        if (isset($this->request->get['limit'])) {
            $url .= '&limit=' . $this->request->get['limit'];
        }

        $this->data['categories'] = array();

        $results = $this->model_catalog_category->getCategories($category_id);

        foreach ($results as $result) {
            $product_total = $this->model_catalog_product->getTotalProducts
(array('filter_category_id' => $result['category_id']));

            $this->data['categories'][] = array(
                'name' => $result['name'] . ' (' . $product_total . ')',
                'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . '_' . $result['category_id'] . $url)
            );
        }

        $this->data['products'] = array();

        $data = array(
            'filter_category_id' => $category_id,
            'sort'               => $sort,
            'order'              => $order,
            'start'              => ($page - 1) * $limit,
            'limit'              => $limit
        );

        $product_total = $this->model_catalog_product->getTotalProducts($data);

        $results = $this->model_catalog_product->getProducts($data);

        foreach ($results as $result) {
            if ($result['image']) {
                $image = $this->model_tool_image->resize($result['image'],
$this->config->get('config_image_product_width'), $this->config->get('config_image_
```

```
product_height'));
                } else {
                    $image = $this->model_tool_image->resize('no_image.jpg', $this->
config->get('config_image_product_width'), $this->config->get('config_image_product_
height'));
                }

                if (($this->config->get('config_customer_price') && $this->customer->
isLogged()) || !$this->config->get('config_customer_price')) {
                    $price = $this->currency->format($this->tax->calculate
($result['price'], $result['tax_class_id'], $this->config->get('config_tax')));
                } else {
                    $price = false;
                }

                if ((float)$result['special']) {
                    $special = $this->currency->format($this->tax->calculate
($result['special'], $result['tax_class_id'], $this->config->get('config_tax')));
                } else {
                    $special = false;
                }

                if ($this->config->get('config_tax')) {
                    $tax = $this->currency->format((float)$result['special'] ?
$result['special'] : $result['price']);
                } else {
                    $tax = false;
                }

                if ($this->config->get('config_review_status')) {
                    $rating = (int)$result['rating'];
                } else {
                    $rating = false;
                }

                $this->data['products'][] = array(
                    'product_id' => $result['product_id'],
                    'thumb'      => $image,
                    'name'       => $result['name'],
                    'description' => truncate(strip_tags(html_entity_decode($result
['description'], ENT_QUOTES, 'UTF-8')), 200) . '..',
                    'price'      => $price,
                    'special'    => $special,
                    'tax'        => $tax,
                    'rating'     => $result['rating'],
                    'reviews'    => sprintf($this->language->get('text_reviews'),
(int)$result['reviews']),
                    'href'       => $this->url->link('product/product', 'path=' .
$this->request->get['path'] . '&product_id=' . $result['product_id'])
                );
            }

            $url = '';
```

```
                if (isset( $ this -> request -> get['limit'])) {
                    $ url . = '&limit = ' . $ this -> request -> get['limit'];
                }

                $ this -> data['sorts'] = array();

                $ this -> data['sorts'][ ] = array(
                    'text' => $ this -> language -> get('text_default'),
                    'value' => 'p.sort_order - ASC',
                    'href' => $ this -> url -> link('product/category', 'path = ' . $ this ->
request -> get['path'] . '&sort = p.sort_order&order = ASC' . $ url)
                );

                $ this -> data['sorts'][ ] = array(
                    'text' => $ this -> language -> get('text_name_asc'),
                    'value' => 'pd.name - ASC',
                    'href' => $ this -> url -> link('product/category', 'path = ' . $ this ->
request -> get['path'] . '&sort = pd.name&order = ASC' . $ url)
                );

                $ this -> data['sorts'][ ] = array(
                    'text' => $ this -> language -> get('text_name_desc'),
                    'value' => 'pd.name - DESC',
                    'href' => $ this -> url -> link('product/category', 'path = ' . $ this ->
request -> get['path'] . '&sort = pd.name&order = DESC' . $ url)
                );

                $ this -> data['sorts'][ ] = array(
                    'text' => $ this -> language -> get('text_price_asc'),
                    'value' => 'p.price - ASC',
                    'href' => $ this -> url -> link('product/category', 'path = ' . $ this ->
request -> get['path'] . '&sort = p.price&order = ASC' . $ url)
                );

                $ this -> data['sorts'][ ] = array(
                    'text' => $ this -> language -> get('text_price_desc'),
                    'value' => 'p.price - DESC',
                    'href' => $ this -> url -> link('product/category', 'path = ' . $ this ->
request -> get['path'] . '&sort = p.price&order = DESC' . $ url)
                );

                $ this -> data['sorts'][ ] = array(
                    'text' => $ this -> language -> get('text_rating_desc'),
                    'value' => 'rating - DESC',
                    'href' => $ this -> url -> link('product/category', 'path = ' . $ this ->
request -> get['path'] . '&sort = rating&order = DESC' . $ url)
                );

                $ this -> data['sorts'][ ] = array(
                    'text' => $ this -> language -> get('text_rating_asc'),
                    'value' => 'rating - ASC',
                    'href' => $ this -> url -> link('product/category', 'path = ' . $ this ->
request -> get['path'] . '&sort = rating&order = ASC' . $ url)
                );
```

```
$this->data['sorts'][] = array(
    'text' => $this->language->get('text_model_asc'),
    'value' => 'p.model-ASC',
    'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . '&sort=p.model&order=ASC' . $url)
);

$this->data['sorts'][] = array(
    'text' => $this->language->get('text_model_desc'),
    'value' => 'p.model-DESC',
    'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . '&sort=p.model&order=DESC' . $url)
);

$url = '';

if (isset($this->request->get['sort'])) {
    $url .= '&sort=' . $this->request->get['sort'];
}

if (isset($this->request->get['order'])) {
    $url .= '&order=' . $this->request->get['order'];
}

$this->data['limits'] = array();

$this->data['limits'][] = array(
    'text' => $this->config->get('config_catalog_limit'),
    'value' => $this->config->get('config_catalog_limit'),
    'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . $url . '&limit=' . $this->config->get('config_catalog_limit'))
);

$this->data['limits'][] = array(
    'text' => 25,
    'value' => 25,
    'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . $url . '&limit=25')
);

$this->data['limits'][] = array(
    'text' => 50,
    'value' => 50,
    'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . $url . '&limit=50')
);

$this->data['limits'][] = array(
    'text' => 75,
    'value' => 75,
    'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . $url . '&limit=75')
);
```

```
        $this->data['limits'][] = array(
            'text' => 100,
            'value' => 100,
            'href' => $this->url->link('product/category', 'path=' . $this->
request->get['path'] . $url . '&limit=100')
        );

        $url = '';

        if (isset($this->request->get['sort'])) {
            $url .= '&sort=' . $this->request->get['sort'];
        }

        if (isset($this->request->get['order'])) {
            $url .= '&order=' . $this->request->get['order'];
        }

        if (isset($this->request->get['limit'])) {
            $url .= '&limit=' . $this->request->get['limit'];
        }

        $pagination = new Pagination();
        $pagination->total = $product_total;
        $pagination->page = $page;
        $pagination->limit = $limit;
        $pagination->text = $this->language->get('text_pagination');
        $pagination->url = $this->url->link('product/category', 'path=' . $this->
request->get['path'] . $url . '&page={page}');

        $this->data['pagination'] = $pagination->render();

        $this->data['sort'] = $sort;
        $this->data['order'] = $order;
        $this->data['limit'] = $limit;

        $this->data['continue'] = $this->url->link('common/home');

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
product/category.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
product/category.tpl';
        } else {
            $this->template = 'default/template/product/category.tpl';
        }

        $this->children = array(
            'common/column_left',
            'common/column_right',
            'common/content_top',
            'common/content_bottom',
            'common/footer',
            'common/header'
        );
```

```
            $this->response->setOutput($this->render());
    } else {
            $url = '';

            if (isset($this->request->get['path'])) {
                $url .= '&path=' . $this->request->get['path'];
            }

            if (isset($this->request->get['sort'])) {
                $url .= '&sort=' . $this->request->get['sort'];
            }

            if (isset($this->request->get['order'])) {
                $url .= '&order=' . $this->request->get['order'];
            }

            if (isset($this->request->get['page'])) {
                $url .= '&page=' . $this->request->get['page'];
            }

            if (isset($this->request->get['limit'])) {
                $url .= '&limit=' . $this->request->get['limit'];
            }

            $this->data['breadcrumbs'][] = array(
                'text'      => $this->language->get('text_error'),
                'href'      => $this->url->link('product/category', $url),
                'separator' => $this->language->get('text_separator')
            );

            $this->document->setTitle($this->language->get('text_error'));

        $this->data['heading_title'] = $this->language->get('text_error');

        $this->data['text_error'] = $this->language->get('text_error');

        $this->data['button_continue'] = $this->language->get('button_continue');

        $this->data['continue'] = $this->url->link('common/home');

            if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
error/not_found.tpl')) {
                $this->template = $this->config->get('config_template') . '/template/
error/not_found.tpl';
            } else {
                $this->template = 'default/template/error/not_found.tpl';
            }

            $this->children = array(
                'common/column_left',
                'common/column_right',
                'common/content_top',
                'common/content_bottom',
```

```
                'common/footer',
                'common/header'
            );

            $ this -> response -> setOutput( $ this -> render());
        }
    }
}
?>
```

浏览商品页面效果如图 12.5 所示。



图 12.5　浏览商品页面效果

12.2.4　购物车

购物车的核心功能由 cart.php 完成，用户可以将自己喜欢的商品暂存到购物车中，最后下订单完成支付。

【例 12.4】　cart.php 的源代码。

```
<?php
class ControllerCheckoutCart extends Controller {
    public function index() {
        $ this -> load_language('checkout/cart');

        if ( $ this -> request -> server['REQUEST_METHOD'] == 'POST') {
```

```
        if (isset($this->request->post['quantity'])) {
                if (!is_array($this->request->post['quantity'])) {
                    if (isset($this->request->post['option'])) {
                        $option = $this->request->post['option'];
                    } else {
                        $option = array();
                    }

                $this->cart->add($this->request->post['product_id'], $this->
request->post['quantity'], $option);
                } else {
                    foreach ($this->request->post['quantity'] as $key => $value) {
                    $this->cart->update($key, $value);
                    }
                }
        }

        if (isset($this->request->post['remove'])) {
            foreach ($this->request->post['remove'] as $key) {
            $this->cart->remove($key);
                }
        }

        if (isset($this->request->post['voucher']) && $this->request->post['voucher']) {
            foreach ($this->request->post['voucher'] as $key) {
            if (isset($this->session->data['vouchers'][$key])) {
                        unset($this->session->data['vouchers'][$key]);
                    }
                }
        }

            if (isset($this->request->post['redirect'])) {
                $this->session->data['redirect'] = $this->request->post['redirect'];
            }

            if (isset($this->request->post['quantity']) || isset($this->request->
post['remove']) || isset($this->request->post['voucher'])) {
                unset($this->session->data['shipping_methods']);
                unset($this->session->data['shipping_method']);
                unset($this->session->data['payment_methods']);
                unset($this->session->data['payment_method']);
                unset($this->session->data['reward']);

                $this->redirect($this->url->link('checkout/cart'));
            }
    }

        if (isset($this->request->get['remove'])) {
        $this->cart->remove($this->request->get['remove']);

        $this->redirect($this->url->link('checkout/cart'));
```

```
}

$this->document->setTitle($this->language->get('heading_title'));

$this->data['breadcrumbs'] = array();

$this->data['breadcrumbs'][] = array(
'href'      => $this->url->link('common/home'),
'text'      => $this->language->get('text_home'),
'separator' => false
);

$this->data['breadcrumbs'][] = array(
'href'      => $this->url->link('checkout/cart'),
'text'      => $this->language->get('heading_title'),
'separator' => $this->language->get('text_separator')
);

if ($this->cart->hasProducts() || (isset($this->session->data['vouchers']) &&
$this->session->data['vouchers'])) {
    $this->data['heading_title'] = $this->language->get('heading_title');

        $this->data['text_select'] = $this->language->get('text_select');
        $this->data['text_weight'] = $this->language->get('text_weight');

    $this->data['column_remove'] = $this->language->get('column_remove');
    $this->data['column_image'] = $this->language->get('column_image');
    $this->data['column_name'] = $this->language->get('column_name');
    $this->data['column_model'] = $this->language->get('column_model');
    $this->data['column_quantity'] = $this->language->get('column_quantity');
        $this->data['column_price'] = $this->language->get('column_price');
    $this->data['column_total'] = $this->language->get('column_total');

    $this->data['button_update'] = $this->language->get('button_update');
    $this->data['button_shopping'] = $this->language->get('button_shopping');
    $this->data['button_checkout'] = $this->language->get('button_checkout');

        if ($this->config->get('config_customer_price') && !$this->customer->
isLogged()) {
            $this->data['attention'] = sprintf($this->language->get('text_login'),
$this->url->link('account/login'), $this->url->link('account/register'));
        } else {
            $this->data['attention'] = '';
        }

        $this->data['action'] = $this->url->link('checkout/cart');

        if ($this->config->get('config_cart_weight')) {
            $this->data['weight'] = $this->weight->format($this->cart->
```

```
getWeight(), $this->config->get('config_weight_class_id'), $this->language->get
('decimal_point'), $this->language->get('thousand_point'));
            } else {
                $this->data['weight'] = false;
            }

            $this->load->model('tool/image');

        $this->data['products'] = array();

        foreach ($this->cart->getProducts() as $result) {
                if ($result['image']) {
                     $image = $this->model_tool_image->resize($result['image'],
$this->config->get('config_image_cart_width'), $this->config->get('config_image_cart
_height'));
                } else {
                    $image = '';
                }

                $option_data = array();

                if ($result['minimum'] > $result['quantity']) {
                    $this->session->data['error'] = sprintf($this->language->get('
error_minimum'), $result['name'], $result['minimum']);
                }

        foreach ($result['option'] as $option) {
                    if ($option['type'] != 'file') {
                        $option_data[] = array(
                            'name' => $option['name'],
                             'value' => (strlen($option['option_value']) > 20 ? substr
($option['option_value'], 0, 20) . '..' : $option['option_value'])
                        );
                    } else {
                        $this->load->library('encryption');

                         $encryption = new Encryption($this->config->get('config_
encryption'));

                        $file = substr($encryption->decrypt($option['option_value']), 0,
strrpos($encryption->decrypt($option['option_value']), '.'));

                        $option_data[] = array(
                            'name' => $option['name'],
                            'value' => (strlen($file) > 20 ? substr($file, 0, 20) . '..' :
$file)
                        );
                    }
        }

                if (($this->config->get('config_customer_price') && $this->customer->
isLogged()) || !$this->config->get('config_customer_price')) {
                      $price = $this->currency->format($this->tax->calculate
($result['price'], $result['tax_class_id'], $this->config->get('config_tax')));
```

```
                } else {
                    $ price = false;
                }

                if (( $ this -> config -> get('config_customer_price') && $ this -> customer ->
isLogged()) || ! $ this -> config -> get('config_customer_price')) {
                    $ total = $ this -> currency -> format( $ this -> tax -> calculate
( $ result['total'], $ result['tax_class_id'], $ this -> config -> get('config_tax')));
                } else {
                    $ total = false;
                }

        $ this -> data['products'][] = array(
            'key'       => $ result['key'],
            'thumb'     => $ image,
                'name'      => $ result['name'],
            'model'     => $ result['model'],
            'option'    => $ option_data,
            'quantity' => $ result['quantity'],
            'stock'     => $ result['stock'],
                'points'    => ( $ result['points'] ? sprintf( $ this -> language -> get
('text_points'), $ result['points']) : ''),
                'price'     => $ price,
                'total'     => $ total,
            'remove'    => $ this -> url -> link('checkout/cart', 'remove = '. $ result['key']),
                'href'      => $ this -> url -> link('product/product', 'product_id = '.
$ result['product_id'])
        );
        }

            // Gift Voucher
            $ this -> data['vouchers'] = array();

            if (isset( $ this -> session -> data['vouchers']) && $ this -> session -> data
['vouchers']) {
                foreach ( $ this -> session -> data['vouchers'] as $ key => $ voucher) {
                    $ this -> data['vouchers'][] = array(
                        'key'          => $ key,
                        'description' => $ voucher['description'],
                        'amount'       => $ this -> currency -> format( $ voucher['amount'])
                    );
                }
            }

            $ total_data = array();
            $ total = 0;
            $ taxes = $ this -> cart -> getTaxes();

            if (( $ this -> config -> get('config_customer_price') && $ this -> customer ->
isLogged()) || ! $ this -> config -> get('config_customer_price')) {

                $ this -> load -> model('setting/extension');

                $ sort_order = array();
```

```
            $results = $this->model_setting_extension->getExtensions('total');

            foreach ($results as $key => $value) {
                $sort_order[$key] = $this->config->get($value['code'] . '_sort_
order');
            }

            array_multisort($sort_order, SORT_ASC, $results);

            foreach ($results as $result) {
                if ($this->config->get($result['code'] . '_status')) {
                    $this->load->model('total/' . $result['code']);

                    $this->{'model_total_' . $result['code']}->getTotal($total_
data, $total, $taxes);
                }
            }

            $sort_order = array();

            foreach ($total_data as $key => $value) {
                $sort_order[$key] = $value['sort_order'];
            }

            array_multisort($sort_order, SORT_ASC, $total_data);
        }

        $this->data['totals'] = $total_data;

        // Modules
        $this->data['modules'] = array();

        if (isset($results)) {
            foreach ($results as $result) {
                if ($this->config->get($result['code'] . '_status') && file_exists
(DIR_APPLICATION . 'controller/total/' . $result['code'] . '.php')) {
                    $this->data['modules'][] = $this->getChild('total/' . $result
['code']);
                }
            }
        }

        if (isset($this->session->data['redirect'])) {
        $this->data['continue'] = $this->session->data['redirect'];

            unset($this->session->data['redirect']);
        } else {
            $this->data['continue'] = $this->url->link('common/home');
        }

    if (!$this->cart->hasStock() && (!$this->config->get('config_stock_checkout') ||
$this->config->get('config_stock_warning'))) {
        $this->data['error_warning'] = $this->language->get('error_stock');
        } elseif (isset($this->session->data['error'])) {
```

```
                $this->data['error_warning'] = $this->session->data['error'];

                unset($this->session->data['error']);
            } else {
                $this->data['error_warning'] = '';
            }

        if (isset($this->session->data['success'])) {
                $this->data['success'] = $this->session->data['success'];

                unset($this->session->data['success']);
            } else {
                $this->data['success'] = '';
            }

            $this->data['checkout'] = $this->url->link('checkout/checkout', '', 'SSL');

            if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
checkout/cart.tpl')) {
                $this->template = $this->config->get('config_template') . '/template/
checkout/cart.tpl';
            } else {
                $this->template = 'default/template/checkout/cart.tpl';
            }

            $this->children = array(
                'common/column_left',
                'common/column_right',
                'common/content_top',
                'common/content_bottom',
                'common/footer',
                'common/header'
            );

            $this->response->setOutput($this->render());
    } else {
        $this->data['heading_title'] = $this->language->get('heading_title');

        $this->data['text_error'] = $this->language->get('text_empty');

        $this->data['button_continue'] = $this->language->get('button_continue');

        $this->data['continue'] = $this->url->link('common/home');

            if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
error/not_found.tpl')) {
                $this->template = $this->config->get('config_template') . '/template/
error/not_found.tpl';
            } else {
                $this->template = 'default/template/error/not_found.tpl';
            }

            $this->children = array(
                'common/column_left',
```

```
                    'common/column_right',
                    'common/content_top',
                    'common/content_bottom',
                    'common/footer',
                    'common/header'
                );

                $this->response->setOutput($this->render());
        }
        }

        public function update() {
            $this->language->load('checkout/cart');

            $json = array();

            if (isset($this->request->post['product_id'])) {
                $this->load->model('catalog/product');

                $product_info = $this->model_catalog_product->getProduct($this->
request->post['product_id']);

                if ($product_info) {
                    // Minimum quantity validation
                    if (isset($this->request->post['quantity'])) {
                        $quantity = $this->request->post['quantity'];
                    } else {
                        $quantity = 1;
                    }

                    $product_total = 0;

                    foreach ($this->session->data['cart'] as $key => $value) {
                        $product = explode(':', $key);

                        if ($product[0] == $this->request->post['product_id']) {
                            $product_total += $value;
                        }
                    }

                    if ($product_info['minimum'] > ($product_total + $quantity)) {
                        $json['error']['warning'] = sprintf($this->language->get('error_
minimum'), $product_info['name'], $product_info['minimum']);
                    }

                    // Option validation
                    if (isset($this->request->post['option'])) {
                        $option = array_filter($this->request->post['option']);
                    } else {
                        $option = array();
                    }

                    $product_options = $this->model_catalog_product->getProductOptions
($this->request->post['product_id']);
```

```
                foreach ( $ product_options as $ product_option) {
                    if ( $ product_option['required'] && (! isset( $ this -> request -> post
['option'][ $ product_option['product_option_id']]) || ! $ this -> request -> post['option']
[ $ product_option['product_option_id']])) {
                        $ json['error'][ $ product_option['product_option_id']] = sprintf
( $ this -> language -> get('error_required'), $ product_option['name']);
                    }
                }
            }

            if (! isset( $ json['error'])) {
                $ this -> cart -> add( $ this -> request -> post['product_id'], $ quantity,
$ option);

                $ json['success'] = sprintf( $ this -> language -> get('text_success'),
$ this -> url -> link('product/product', 'product_id = ' . $ this -> request -> post['product_
id']), $ product_info['name'], $ this -> url -> link('checkout/cart'));

                unset( $ this -> session -> data['shipping_methods']);
                unset( $ this -> session -> data['shipping_method']);
                unset( $ this -> session -> data['payment_methods']);
                unset( $ this -> session -> data['payment_method']);
            } else {
                $ json['redirect'] = str_replace('&amp;', '&', $ this -> url -> link('product/
product', 'product_id = ' . $ this -> request -> post['product_id']));
            }
        }

    if (isset( $ this -> request -> post['remove'])) {
     $ this -> cart -> remove( $ this -> request -> post['remove']);
    }

    if (isset( $ this -> request -> post['voucher'])) {
            if ( $ this -> session -> data['vouchers'][ $ this -> request -> post['voucher']]) {
                unset( $ this -> session -> data['vouchers'][ $ this -> request -> post
['voucher']]);
            }
        }

        $ this -> load -> model('tool/image');

        $ this -> data['text_empty'] = $ this -> language -> get('text_empty');

        $ this -> data['button_checkout'] = $ this -> language -> get('button_checkout');
        $ this -> data['button_remove'] = $ this -> language -> get('button_remove');

        $ this -> data['products'] = array();

        foreach ( $ this -> cart -> getProducts() as $ result) {
            if ( $ result['image']) {
                $ image = $ this -> model_tool_image -> resize( $ result['image'], 40, 40);
            } else {
                $ image = '';
```

```
        }

        $option_data = array();

        foreach ($result['option'] as $option) {
            if ($option['type'] != 'file') {
                $option_data[] = array(
                    'name' => $option['name'],
                    'value' => (strlen($option['option_value']) > 20 ? substr($option
['option_value'], 0, 20) . '..' : $option['option_value'])
                );
            } else {
                $this->load->library('encryption');

                    $encryption = new Encryption($this->config->get('config_
encryption'));

                $file = substr($encryption->decrypt($option['option_value']), 0,
strrpos($encryption->decrypt($option['option_value']), '.'));

                $option_data[] = array(
                    'name' => $option['name'],
                     'value' => (strlen($file) > 20 ? substr($file, 0, 20) . '..' :
$file)
                );
            }
        }

        if (($this->config->get('config_customer_price') && $this->customer->
isLogged()) || !$this->config->get('config_customer_price')) {
            $price = $this->currency->format($this->tax->calculate($result
['price'], $result['tax_class_id'], $this->config->get('config_tax')));
        } else {
            $price = false;
        }

        if (($this->config->get('config_customer_price') && $this->customer->
isLogged()) || !$this->config->get('config_customer_price')) {
            $total = $this->currency->format($this->tax->calculate($result
['total'], $result['tax_class_id'], $this->config->get('config_tax')));
        } else {
            $total = false;
        }

        $this->data['products'][] = array(
            'key'      => $result['key'],
            'thumb'    => $image,
            'name'     => $result['name'],
            'model'    => $result['model'],
            'option'   => $option_data,
            'quantity' => $result['quantity'],
            'stock'    => $result['stock'],
            'price'    => $price,
            'total'    => $total,
```

```
                    'href'     => $this->url->link('product/product', 'product_id=' .
$result['product_id'])
                );
            }

            // Gift Voucher
            $this->data['vouchers'] = array();

            if (isset($this->session->data['vouchers']) && $this->session->data
['vouchers']){
                foreach ($this->session->data['vouchers'] as $key => $voucher) {
                    $this->data['vouchers'][] = array(
                        'key'         => $key,
                        'description' => $voucher['description'],
                        'amount'      => $this->currency->format($voucher['amount'])
                    );
                }
            }

            // Calculate Totals
            $total_data = array();
            $total = 0;
            $taxes = $this->cart->getTaxes();

            if (($this->config->get('config_customer_price') && $this->customer->
isLogged()) || !$this->config->get('config_customer_price')) {
                $this->load->model('setting/extension');

                $sort_order = array();

                $results = $this->model_setting_extension->getExtensions('total');

                foreach ($results as $key => $value) {
                    $sort_order[$key] = $this->config->get($value['code'] . '_sort_order
');
                }

                array_multisort($sort_order, SORT_ASC, $results);

                foreach ($results as $result) {
                    if ($this->config->get($result['code'] . '_status')) {
                        $this->load->model('total/' . $result['code']);

                        $this->{'model_total_' . $result['code']}->getTotal($total_data,
$total, $taxes);
                    }
                }

                $sort_order = array();

                foreach ($total_data as $key => $value) {
                    $sort_order[$key] = $value['sort_order'];
                }
```

```
            array_multisort( $ sort_order, SORT_ASC, $ total_data);
        }

        $ json['total'] = sprintf( $ this->language->get('text_items'), $ this->cart->
countProducts() + (isset( $ this->session->data['vouchers']) ? count( $ this->session->
data['vouchers']) : 0), $ this->currency->format( $ total));

        $ this->data['totals'] = $ total_data;

        $ this->data['checkout'] = $ this->url->link('checkout/checkout', '', 'SSL');

        if (file_exists(DIR_TEMPLATE . $ this->config->get('config_template') . '/template/
common/cart.tpl')) {
            $ this->template = $ this->config->get('config_template') . '/template/
common/cart.tpl';
        } else {
            $ this->template = 'default/template/common/cart.tpl';
        }

        $ json['output'] = $ this->render();

        $ this->load->library('json');

        $ this->response->setOutput(Json::encode( $ json));
    }
}
?>
```

购物车页面效果如图 12.6 所示。



图 12.6　购物车页面效果

12.2.5 下单

所谓下单即完成商品购买的过程,系统会自动创建相应的订单页面,用户确认所需购买的商品同时填写送货地址等信息,确认并进行支付后即可完成商品购买。此功能由checkout. php、address. php、payment. php、confirmation. php、process. php 和 success. php等多个 PHP 页面组成,其中,checkout. php 主要用于触发购买动作,address. php 供用户输入送货地址信息(或直接选取常用地址信息),payment. php 页面用于输入支付信息,confirmation. php 用于最后购买前的确认,确认后所有信息提交给 process. php 进行处理,如果处理成功则返回 success. php 给用户。

【例 12.5】 checkout. php 的源代码。

```
<?php
class ControllerCheckoutCheckout extends Controller {
    private $direct_payments = array('cod','cheque','free_checkout','bank_transfer');

    public function index() {
        if (!$this->customer->isLogged()) {
            $this->session->data['redirect'] = $this->url->link('checkout/checkout', '',
'SSL');

            $this->redirect($this->url->link('account/login', '', 'SSL'));
        }

        if ((!$this->cart->hasProducts() && (!isset($this->session->data['vouchers']) ||
!$this->session->data['vouchers'])) || (!$this->cart->hasStock() && !$this->config->
get('config_stock_checkout'))) {
            $this->redirect($this->url->link('checkout/cart'));
        }

        if(!isset($this->session->data['checkout_token']))
        $this->session->data['checkout_token'] = md5(mt_rand());

        $this->log->debug('IlexDebug:: Begin Checkout : '.$this->session->data
['checkout_token']);

        $this->load_language('checkout/checkout');

        $this->document->setTitle($this->language->get('heading_title'));

        // Minimum quantity validation
        $products = $this->cart->getProducts();

        foreach ($products as $product) {
            $product_total = 0;

            foreach ($this->session->data['cart'] as $key => $quantity) {
```

```
                $product_2 = explode(':', $key);

                if ($product_2[0] == $product['product_id']) {
                    $product_total += $quantity;
                }
            }

            if ($product['minimum'] > $product_total) {
                $this->session->data['error'] = sprintf($this->language->get
('error_minimum'), $product['name'], $product['minimum']);

                $this->redirect($this->url->link('checkout/cart'));
            }
        }

        $this->data['breadcrumbs'] = array();

        $this->data['breadcrumbs'][] = array(
    'text' => $this->language->get('text_home'),
            'href'      => $this->url->link('common/home'),
    'separator' => false
        );

        $this->data['breadcrumbs'][] = array(
           'text'      => $this->language->get('text_cart'),
           'href'      => $this->url->link('checkout/cart'),
           'separator' => $this->language->get('text_separator')
        );

        $this->data['breadcrumbs'][] = array(
           'text'      => $this->language->get('heading_title'),
           'href'      => $this->url->link('checkout/checkout', '', 'SSL'),
           'separator' => $this->language->get('text_separator')
        );

        $this->data['cart'] = $this->url->link('checkout/cart');
        $this->data['logged'] = $this->customer->isLogged();
        $this->data['shipping_required'] = $this->cart->hasShipping();
        $this->data['has_address'] = 1;

        if($this->customer->getAddressId()){
            $this->data['shipping_address'] = $this->shipping_address();
            $this->data['shipping_methods'] = $this->shipping_method();
        }else{
            $this->data['has_address'] = 0;
            $this->data['shipping_address'] = $this->shipping_new_address();
            $this->data['shipping_methods'] = $this->default_shipping_method();
        }
```

```
$this->data['payment_methods'] = $this->payment_method();

$this->data['comment'] = '';
if (isset($this->session->data['comment'])) {
    $this->data['comment'] = $this->session->data['comment'];
}

$results = $this->model_setting_extension->getExtensions('total');

// Modules
$this->data['modules'] = array();

if (isset($results)) {
    foreach ($results as $result) {
        if ($this->config->get($result['code'] . '_status') && file_exists(DIR_
APPLICATION . 'controller/total/' . $result['code'] . '.php')) {
            $this->data['modules'][] = $this->getChild('total/' . $result
['code']);
        }
    }
}

$this->data['modules'][] = $this->getChild('total/comment');

$this->data['order_confirm'] = $this->confirm();

$this->data['payment']   = $this->payment();

$total_data = array();
$total = 0;
$taxes = $this->cart->getTaxes();

$this->load->model('setting/extension');

$sort_order = array();

$results = $this->model_setting_extension->getExtensions('total');

foreach ($results as $key => $value) {
    $sort_order[$key] = $this->config->get($value['code'] . '_sort_order');
}

array_multisort($sort_order, SORT_ASC, $results);

foreach ($results as $result) {
    if ($this->config->get($result['code'] . '_status')) {
        $this->load->model('total/' . $result['code']);
```

```
                $this->{'model_total_'. $result['code']}->getTotal($total_data,
$total, $taxes);
            }
        }

        $sort_order = array();

        foreach ($total_data as $key => $value) {
            $sort_order[$key] = $value['sort_order'];
        }

        array_multisort($sort_order, SORT_ASC, $total_data);

        $this->data['totals'] = $total_data;

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
checkout/checkout.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
checkout/checkout.tpl';
        } else {
            $this->template = 'default/template/checkout/checkout.tpl';
        }

        $this->children = array(
            'common/column_left',
            'common/column_right',
            'common/content_top',
            'common/content_bottom',
            'common/footer',
            'common/checkout_header'
        );

        $this->response->setOutput($this->render());
    }


    public function shipping_address() {
        $this->load_language('checkout/checkout');

        $this->load->model('account/address');

        $this->data['type'] = 'shipping';

        if (isset($this->session->data['shipping_address_id'])) {
            $this->data['address_id'] = $this->session->data['shipping_address_id'];
             $this->log->debug('IlexDebug:: session : '. $this->session->data
['shipping_address_id']);
        } else {
```

```
            $this->data['address_id'] = $this->customer->getAddressId();
            $this->session->data['shipping_address_id'] = $this->data['address_id'];
            $this->log->debug('IlexDebug:: default address_id : '. $this->session->
data['shipping_address_id']);
        }

        $this->data['addresses'] = $this->model_account_address->getAddresses();

        $this->data['country_id'] = $this->config->get('config_country_id');

        $this->load->model('localisation/country');

        $this->data['countries'] = $this->model_localisation_country->getCountries();

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
checkout/address.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
checkout/address.tpl';
        } else {
            $this->template = 'default/template/checkout/address.tpl';
        }

        return $this->render();
    }

    private function shipping_new_address() {
        $this->load_language('checkout/checkout');

        $this->load->model('account/address');

        $this->data['type'] = 'shipping';

        $this->data['country_id'] = $this->config->get('config_country_id');

        $this->load->model('localisation/country');

        $this->data['countries'] = $this->model_localisation_country->getCountries();

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
checkout/address_new.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
checkout/address_new.tpl';
        } else {
            $this->template = 'default/template/checkout/address_new.tpl';
        }

        return $this->render();
    }
```

```
public function shipping_method() {
    $this->load_language('checkout/checkout');

    $this->load->model('account/address');

    if(isset($this->session->data['shipping_address_id']))
        $shipping_address = $this->model_account_address->getAddress($this->
session->data['shipping_address_id']);
    else
        $shipping_address = $this->model_account_address->getAddress($this->
customer->getAddressId());

    $this->tax->setZone($shipping_address['country_id'], $shipping_address['zone_
id']);

    if (!isset($this->session->data['shipping_methods'])) {
        $quote_data = array();

        $this->load->model('setting/extension');

        $results = $this->model_setting_extension->getExtensions('shipping');

        foreach ($results as $result) {
            if ($this->config->get($result['code'] . '_status')) {
                $this->load->model('shipping/' . $result['code']);

                $quote = $this->{'model_shipping_' . $result['code']}->getQuote
($shipping_address);
                if ($quote) {
                    $quote_data[$result['code']] = array(
                        'title'         => $quote['title'],
                        'quote'         => $quote['quote'],
                        'description'   => $quote['description'],
                        'sort_order'    => $quote['sort_order'],
                        'error'         => $quote['error']
                    );
                }
            }
        }

        $sort_order = array();

        foreach ($quote_data as $key => $value) {
            $sort_order[$key] = $value['sort_order'];
        }

        array_multisort($sort_order, SORT_ASC, $quote_data);
```

```
            $this->session->data['shipping_methods'] = $quote_data;

        }

        if (isset($this->session->data['shipping_methods']) && !$this->session->data
['shipping_methods']) {
            $this->data['error_warning'] = sprintf($this->language->get('error_no_
shipping'), $this->url->link('information/contact'));
        } else {
            $this->data['error_warning'] = '';
        }

        if (isset($this->session->data['shipping_methods'])) {
            $this->data['shipping_methods'] = $this->session->data['shipping_methods'];
        } else {
            $this->data['shipping_methods'] = array();
        }

        if (isset($this->session->data['shipping_method']['code'])) {
            $this->data['shipping_code'] = $this->session->data['shipping_method']
['code'];
            $this->customer->setShippingMethod($this->session->data['shipping_
method']['code']);
        } else {
            $this->data['shipping_code'] = $this->customer->getShippingMethod();

            $shipping = explode('.', $this->data['shipping_code']);
            if(isset($shipping[0])&&isset($shipping[1])){
                if(isset($this->session->data['shipping_methods'][$shipping[0]]
['quote'][$shipping[1]]))
                    $this->session->data['shipping_method'] = $this->session->data
['shipping_methods'][$shipping[0]]['quote'][$shipping[1]];
                else
                    $this->data['shipping_code'] = '';
            }
        }

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
checkout/shipping.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
checkout/shipping.tpl';
        } else {
            $this->template = 'default/template/checkout/shipping.tpl';
        }

        return $this->render();
    }
```

```
public function default_shipping_method() {
    $ this -> load_language('checkout/checkout');

    $ shipping_address['country_id'] = $ this -> config -> get('config_country_id');
    $ shipping_address['zone_id'] = $ this -> config -> get('config_zone_id');
    $ this -> tax -> setZone( $ shipping_address['country_id'], $ shipping_address['zone_
id']);

    $ this -> tax -> setZone( $ shipping_address['country_id'], $ shipping_address['zone_
id']);

    $ quote_data = array();

    $ this -> load -> model('setting/extension');

    $ results = $ this -> model_setting_extension -> getExtensions('shipping');

    foreach ( $ results as $ result) {
        if ( $ this -> config -> get( $ result['code'] . '_status')) {
            $ this -> load -> model('shipping/' . $ result['code']);

            $ quote = $ this ->{ 'model_shipping_' . $ result[ 'code' ]} -> getQuote
( $ shipping_address);

            if ( $ quote) {
                $ quote_data[ $ result['code']] = array(
                    'title'         => $ quote['title'],
                    'quote'         => $ quote['quote'],
    'description'   => $ quote['description'],
                    'sort_order'    => $ quote['sort_order'],
                    'error'         => $ quote['error']
                );
            }
        }
    }

    $ sort_order = array();

    foreach ( $ quote_data as $ key => $ value) {
        $ sort_order[ $ key] = $ value['sort_order'];
    }

    array_multisort( $ sort_order, SORT_ASC, $ quote_data);

    $ this -> session -> data['shipping_methods'] = $ quote_data;

    $ this -> data['error_warning'] = '';
```

```
        if (isset( $ this -> session -> data[ 'shipping_methods'])) {
            $ this -> data[ 'shipping_methods']  =  $ this -> session -> data[ 'shipping_methods'];
        } else {
            $ this -> data[ 'shipping_methods']  =  array();
        }

        if (isset( $ this -> session -> data[ 'shipping_method'][ 'code'])) {
            $ this -> data[ 'shipping_code']  =  $ this -> session -> data[ 'shipping_method']
[ 'code'];
            $ this -> customer -> setShippingMethod( $ this -> session -> data[ 'shipping_
method'][ 'code']);
        } else {
            $ this -> data[ 'shipping_code']  =  $ this -> customer -> getShippingMethod();
        }

        if (file_exists(DIR_TEMPLATE . $ this -> config -> get( 'config_template') . '/template/
checkout/shipping.tpl')) {
            $ this -> template  =  $ this -> config -> get( 'config_template') . '/template/
checkout/shipping.tpl';
        } else {
            $ this -> template  =  'default/template/checkout/shipping.tpl';
        }

        return $ this -> render();
    }

    public function payment_method() {
        $ this -> load_language( 'checkout/checkout');

        $ this -> load -> model( 'account/address');

        if(isset( $ this -> session -> data[ 'shipping_address_id']))

        $ this -> session -> data[ 'payment_address_id'] = $ this -> session -> data[ 'shipping_
address_id'];
        else

        $ this -> session -> data[ 'payment_address_id'] = $ this -> customer -> getAddressId();

        $ payment_address  =  $ this -> model_account_address -> getAddress( $ this -> session ->
data[ 'payment_address_id']);

        $ total_data  =  array();
        $ total  =  0;
        $ taxes  =  $ this -> cart -> getTaxes();

        $ this -> load -> model( 'setting/extension');

        $ sort_order  =  array();
```

```
$ results = $ this -> model_setting_extension -> getExtensions('total');

foreach ( $ results as $ key => $ value) {
    $ sort_order[ $ key] = $ this -> config -> get( $ value['code'] . '_sort_order');
}

array_multisort( $ sort_order, SORT_ASC, $ results);

foreach ( $ results as $ result) {
    if ( $ this -> config -> get( $ result['code'] . '_status')) {
        $ this -> load -> model('total/' . $ result['code']);

         $ this ->{ 'model_total_' . $ result[ 'code' ]} -> getTotal( $ total_data,
$ total, $ taxes);
    }
}
// Payment Methods
$ method_data = array();

$ this -> load -> model('setting/extension');

$ results = $ this -> model_setting_extension -> getExtensions('payment');

foreach ( $ results as $ result) {
    if ( $ this -> config -> get( $ result['code'] . '_status')) {
        $ this -> load -> model('payment/' . $ result['code']);

         $ method = $ this ->{ 'model_payment_' . $ result[ 'code' ]} -> getMethod
( $ payment_address, $ total);

        if ( $ method) {
            $ method_data[ $ result['code']] = $ method;
        }
    }
}

$ sort_order = array();

foreach ( $ method_data as $ key => $ value) {
    $ sort_order[ $ key] = $ value['sort_order'];
}

array_multisort( $ sort_order, SORT_ASC, $ method_data);

$ this -> session -> data['payment_methods'] = $ method_data;

if (isset( $ this -> session -> data['payment_methods']) && ! $ this -> session -> data
['payment_methods']) {
```

```
            $this->data['error_warning'] = sprintf($this->language->get('error_no_payment'), $this->url->link('information/contact'));
        } else {
            $this->data['error_warning'] = '';
        }

        if (isset($this->session->data['payment_methods'])) {
            $this->data['payment_methods'] = $this->session->data['payment_methods'];
        } else {
            $this->data['payment_methods'] = array();
        }

        if (isset($this->session->data['payment_method']['code'])) {
            $this->data['payment_code'] = $this->session->data['payment_method']['code'];
            $this->customer->setPaymentMethod($this->data['payment_code']);
            $this->log->debug('IlexDebug:: Checkout payment_method 1 '. $this->data['payment_code']);
        } else if($this->customer->getPaymentMethod()!= ''){
            $this->data['payment_code'] = $this->customer->getPaymentMethod();
            $this->log->debug('IlexDebug:: Checkout payment_method 2 '. $this->data['payment_code']);
        } else{

            $this->data['payment_code'] = $this->config->get('config_default_payment');
            $this->log->debug('IlexDebug:: Checkout payment_method 3 '. $this->data['payment_code']);
        }

        $this->session->data['payment_method']['code'] = $this->data['payment_code'] ;

        if ($this->config->get('config_checkout_id')) {
            $this->load->model('catalog/information');

            $information_info = $this->model_catalog_information->getInformation($this->config->get('config_checkout_id'));

            if ($information_info) {
                $this->data['text_agree'] = sprintf($this->language->get('text_agree'), $this->url->link('information/information/info', 'information_id=' . $this->config->get('config_checkout_id'), 'SSL'), $information_info['title'], $information_info['title']);
            } else {
                $this->data['text_agree'] = '';
            }
        } else {
            $this->data['text_agree'] = '';
        }
```

```
        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
checkout/payment.tpl')) {
            $this->template = $this->config->get('config_template') . '/template/
checkout/payment.tpl';
        } else {
            $this->template = 'default/template/checkout/payment.tpl';
        }
        return $this->render();

    }

    public function payment() {
        if (isset($this->session->data['payment_method'])) {
            $this->data['payment_code'] = $this->session->data['payment_method']
['code'];
            $this->customer->setPaymentMethod($this->data['payment_code']);
            $this->log->debug('IlexDebug:: payment comfirm payment_method '. $this->
data['payment_code']);
        } else if($this->customer->getPaymentMethod()!= ''){
            $this->data['payment_code'] = $this->customer->getPaymentMethod();
            $this->log->debug('IlexDebug:: payment comfirm 2 getPaymentMethod '. $this->
data['payment_code']);
        } else{

            $this->data['payment_code'] = $this->config->get('config_default_payment');
            $this->log->debug('IlexDebug:: payment comfirm 3 '. $this->data['payment_
code']);
        }

        return $this->getChild('payment/' . $this->data['payment_code']);
    }

    public function confirm() {
        if ((! $this->cart->hasProducts() && (! isset($this->session->data['vouchers']) ||
! $this->session->data['vouchers'])) || (! $this->cart->hasStock() && ! $this->config->
get('config_stock_checkout'))) {
            $json['redirect'] = $this->url->link('checkout/cart');
        }

        $this->load->model('account/address');

        if ($this->cart->hasShipping()) {
            $this->load->model('account/address');

            if ($this->customer->isLogged()) {
                if(isset($this->session->data['shipping_address_id']))
                    $shipping_address = $this->model_account_address->getAddress
($this->session->data['shipping_address_id']);
                else
```

```
                    $ shipping_address = $ this->model_account_address->getAddress
( $ this->customer->getAddressId());
            } else{
                $ this->redirect( $ this->url->link('account/login', '', 'SSL'));
            }

            if (!isset( $ shipping_address)) {
                $ json['redirect'] = $ this->url->link('checkout/checkout', '', 'SSL');
            }

            if (!isset( $ this->session->data['shipping_method'])) {
                $ json['redirect'] = $ this->url->link('checkout/checkout', '', 'SSL');
            }
        } else {
            unset( $ this->session->data['guest']['shipping']);
            unset( $ this->session->data['shipping_address_id']);
            unset( $ this->session->data['shipping_method']);
            unset( $ this->session->data['shipping_methods']);
        }

        $ json = array();

        if (! $ json) {
            $ total_data = array();
            $ total = 0;
            $ taxes = $ this->cart->getTaxes();

            $ this->load->model('setting/extension');

            $ sort_order = array();

            $ results = $ this->model_setting_extension->getExtensions('total');

            foreach ( $ results as $ key => $ value) {
                $ sort_order[ $ key] = $ this->config->get( $ value['code'] . '_sort_order
');
            }

            array_multisort( $ sort_order, SORT_ASC, $ results);

            foreach ( $ results as $ result) {
                if ( $ this->config->get( $ result['code'] . '_status')) {
                    $ this->load->model('total/' . $ result['code']);

                    $ this->{'model_total_' . $ result['code']}->getTotal( $ total_data,
 $ total, $ taxes);
                }
            }
```

```
$sort_order = array();

foreach ($total_data as $key => $value) {
    $sort_order[$key] = $value['sort_order'];
    $this->log->debug('IlexDebug:: get total data '. $value['code']);
}

array_multisort($sort_order, SORT_ASC, $total_data);

$this->language->load('checkout/checkout');

$data = array();

$data['invoice_prefix'] = $this->config->get('config_invoice_prefix');
$data['store_id'] = $this->config->get('config_store_id');
$data['store_name'] = $this->config->get('config_name');

if ($data['store_id']) {
    $data['store_url'] = $this->config->get('config_url');
} else {
    $data['store_url'] = HTTP_SERVER;
}

if ($this->customer->isLogged()) {
    $data['customer_id'] = $this->customer->getId();
    $data['customer_group_id'] = $this->customer->getCustomerGroupId();
    $data['firstname'] = $this->customer->getFirstName();
    $data['lastname'] = $this->customer->getLastName();
    $data['email'] = $this->customer->getEmail();
    $data['telephone'] = $this->customer->getTelephone();
    $data['fax'] = $this->customer->getFax();

    $this->load->model('account/address');
    if(isset($this->session->data['payment_address_id'])){
        $payment_address = $this->model_account_address->getAddress
($this->session->data['payment_address_id']);
    }
} elseif (isset($this->session->data['guest'])) {
    $data['customer_id'] = 0;
    $data['customer_group_id'] = $this->config->get('config_customer_
group_id');
    $data['firstname'] = $this->session->data['guest']['firstname'];
    $data['lastname'] = $this->session->data['guest']['lastname'];
    $data['email'] = $this->session->data['guest']['email'];
    $data['telephone'] = $this->session->data['guest']['telephone'];
    $data['fax'] = $this->session->data['guest']['fax'];

    $payment_address = $this->session->data['guest']['payment'];
```

```
            }

            if(isset( $ payment_address)){
                $ data['payment_firstname'] = $ payment_address['firstname'];
                $ data['payment_lastname'] = $ payment_address['lastname'];
                $ data['payment_company'] = $ payment_address['company'];
                $ data['payment_address_1'] = $ payment_address['address_1'];
                $ data['payment_address_2'] = $ payment_address['address_2'];
                $ data['payment_city'] = $ payment_address['city'];
                $ data['payment_postcode'] = $ payment_address['postcode'];
                $ data['payment_zone'] = $ payment_address['zone'];
                $ data['payment_zone_id'] = $ payment_address['zone_id'];
                $ data['payment_country'] = $ payment_address['country'];
                $ data['payment_country_id'] = $ payment_address['country_id'];
                $ data['payment_address_format'] = $ payment_address['address_format'];
            }else{
                $ data['payment_firstname'] = '';
                $ data['payment_lastname'] = '';
                $ data['payment_company'] = '';
                $ data['payment_address_1'] = '';
                $ data['payment_address_2'] = '';
                $ data['payment_city'] = '';
                $ data['payment_postcode'] = '';
                $ data['payment_zone'] = '';
                $ data['payment_zone_id'] = '';
                $ data['payment_country'] = '';
                $ data['payment_country_id'] = '';
                $ data['payment_address_format'] = '';
            }

            if (isset( $ this -> session -> data['payment_method']['title'])) {
                 $ data['payment_method'] = $ this -> session -> data['payment_method']
['title'];
            } else {
                $ data['payment_method'] = $ this -> customer -> getPaymentMethodTitle();
            }

            if ( $ this -> cart -> hasShipping()) {
                if ( $ this -> customer -> isLogged()) {
                    $ this -> load -> model('account/address');
                    if(isset( $ this -> session -> data['shipping_address_id']))
                            $ shipping_address = $ this -> model_account_address ->
getAddress( $ this -> session -> data['shipping_address_id']);
                    else
                            $ shipping_address = $ this -> model_account_address ->
getAddress( $ this -> customer -> getAddressId());

                } else{
```

```
            $ this -> redirect( $ this -> url -> link('account/login', '', 'SSL'));
        }

        $ data['shipping_firstname'] = $ shipping_address['firstname'];
        $ data['shipping_lastname'] = $ shipping_address['lastname'];
        $ data['shipping_mobile'] = $ shipping_address['mobile'];
        $ data['shipping_phone'] = $ shipping_address['phone'];
        $ data['shipping_company'] = $ shipping_address['company'];
        $ data['shipping_address_1'] = $ shipping_address['address_1'];
        $ data['shipping_address_2'] = $ shipping_address['address_2'];
        $ data['shipping_city'] = $ shipping_address['city'];
        $ data['shipping_postcode'] = $ shipping_address['postcode'];
        $ data['shipping_zone'] = $ shipping_address['zone'];
        $ data['shipping_zone_id'] = $ shipping_address['zone_id'];
        $ data['shipping_country'] = $ shipping_address['country'];
        $ data['shipping_country_id'] = $ shipping_address['country_id'];
        $ data['shipping_address_format'] = $ shipping_address['address_format'];

        if (isset( $ this -> session -> data['shipping_method']['title'])) {
            $ data['shipping_method'] = $ this -> session -> data['shipping_method']
['title'];
        } else {
            $ data['shipping_method'] = $ this -> customer -> getShippingMethodTitle();
        }
    } else {
        $ data['shipping_firstname'] = '';
        $ data['shipping_lastname'] = '';
        $ data['shipping_mobile'] = '';
        $ data['shipping_phone'] = '';
        $ data['shipping_company'] = '';
        $ data['shipping_address_1'] = '';
        $ data['shipping_address_2'] = '';
        $ data['shipping_city'] = '';
        $ data['shipping_postcode'] = '';
        $ data['shipping_zone'] = '';
        $ data['shipping_zone_id'] = '';
        $ data['shipping_country'] = '';
        $ data['shipping_country_id'] = '';
        $ data['shipping_address_format'] = '';
        $ data['shipping_method'] = '';
    }

    if ( $ this -> cart -> hasShipping()) {
        $ this -> tax -> setZone( $ shipping_address['country_id'], $ shipping_
address['zone_id']);
    } else {
        $ this -> tax -> setZone( $ payment_address['country_id'], $ payment_address
['zone_id']);
```

```
        }

        $product_data = array();

        foreach ($this->cart->getProducts() as $product) {
            $option_data = array();

            foreach ($product['option'] as $option) {
                if ($option['type'] != 'file') {
                    $option_data[] = array(
                        'product_option_id'       => $option['product_option_
id'],
                        'product_option_value_id' => $option['product_option_
value_id'],
                        'product_option_id'       => $option['product_option_
id'],
                        'product_option_value_id' => $option['product_option_
value_id'],
                        'option_id'             => $option['option_id'],
                        'option_value_id'       => $option['option_value_id'],
                        'name'                  => $option['name'],
                        'value'                 => $option['option_value'],
                        'type'                  => $option['type']
                    );
                } else {
                    $this->load->library('encryption');

                    $encryption = new Encryption($this->config->get('config_
encryption'));

                    $option_data[] = array(
                        'product_option_id'       => $option['product_option_
id'],
                        'product_option_value_id' => $option['product_option_
value_id'],
                        'product_option_id'       => $option['product_option_
id'],
                        'product_option_value_id' => $option['product_option_
value_id'],
                        'option_id'             => $option['option_id'],
                        'option_value_id'       => $option['option_value_id'],
                        'name'                  => $option['name'],
                        'value'                  => $encryption->decrypt
($option['option_value']),
                        'type'                  => $option['type']
                    );
                }
            }
```

```
                    $product_data[] = array(
                        'product_id' => $product['product_id'],
                        'href' => $this->url->link('product/product', '&product_id=' .
$product['product_id']),
                        'name'      => $product['name'],
                        'model'     => $product['model'],
                        'option' => $option_data,
                        'download' => $product['download'],
                        'quantity' => $product['quantity'],
                        'subtract' => $product['subtract'],
                        'price'     => $product['price'],
                        'total'     => $product['total'],
                        'tax'       => $this->tax->getRate($product['tax_class_
id'])
                    );
                }

                // Gift Voucher
                if (isset($this->session->data['vouchers']) && $this->session->data
['vouchers']) {
                    foreach ($this->session->data['vouchers'] as $voucher) {
                        $product_data[] = array(
                            'product_id' => 0,
                            'href' => '#',
                            'name'      => $voucher['description'],
                            'model'     => '',
                            'option' => array(),
                            'download' => array(),
                            'quantity' => 1,
                            'subtract' => false,
                            'price'     => $voucher['amount'],
                            'total'     => $voucher['amount'],
                            'tax'       => 0
                        );
                    }
                }

                $data['products'] = $product_data;
                $this->data['products'] = $product_data;

                $data['totals'] = $total_data;

                if(isset($this->session->data['comment'])){
                    $data['comment'] = $this->session->data['comment'];
                }else{
                    $data['comment'] = '';
                }
                $data['total'] = $total;
```

```
        $data['reward'] = $this->cart->getTotalRewardPoints();

        if (isset($this->request->cookie['tracking'])) {
            $this->load->model('affiliate/affiliate');

            $affiliate_info = $this->model_affiliate_affiliate->getAffiliateByCode
($this->request->cookie['tracking']);

            if ($affiliate_info) {
                $data['affiliate_id'] = $affiliate_info['affiliate_id'];
                $data['commission'] = ($total / 100) * $affiliate_info['commission'];
            } else {
                $data['affiliate_id'] = 0;
                $data['commission'] = 0;
            }
        } else {
            $data['affiliate_id'] = 0;
            $data['commission'] = 0;
        }

        $data['language_id'] = $this->config->get('config_language_id');
        $data['currency_id'] = $this->currency->getId();
        $data['currency_code'] = $this->currency->getCode();
        $data['currency_value'] = $this->currency->getValue($this->currency->
getCode());
        $data['ip'] = $this->request->server['REMOTE_ADDR'];

        $this->load->model('checkout/order');

        if (isset($this->session->data['payment_method'])) {
            $this->data['payment_code'] = $this->session->data['payment_method']
['code'];
            $this->customer->setPaymentMethod($this->data['payment_code']);
            $this->log->debug('IlexDebug:: Checkout comfirm payment_method '. $this->
data['payment_code']);
        } else if($this->customer->getPaymentMethod()!= ''){
            $this->data['payment_code'] = $this->customer->getPaymentMethod();
            $this->log->debug('IlexDebug:: Checkout comfirm 2 getPaymentMethod '.
$this->data['payment_code']);
        } else{

            $this->data['payment_code'] = $this->config->get('config_default_
payment');
            $this->log->debug('IlexDebug:: Checkout comfirm 3 '. $this->data
['payment_code']);
        }

        $data['payment_code'] = $this->data['payment_code'];
```

```
        if(!isset($this->session->data['order_id']))
            $this->session->data['order_id'] = $this->model_checkout_order->
create($data);
        else
            $this->model_checkout_order->modify($data);

        // Gift Voucher
        if (isset($this->session->data['vouchers']) && is_array($this->session->
data['vouchers'])) {
            $this->load->model('checkout/voucher');

            foreach ($this->session->data['vouchers'] as $voucher) {

                $this->model_checkout_voucher->addVoucher($this->session->
data['order_id'], $voucher);
            }
        }

        // Gift Voucher
        $this->data['vouchers'] = array();

        if (isset($this->session->data['vouchers']) && $this->session->data
['vouchers']) {
            foreach ($this->session->data['vouchers'] as $key => $voucher) {
                $this->data['vouchers'][] = array(
                    'description' => $voucher['description'],
                    'amount'      => $this->currency->format($voucher['amount'])
                );
            }
        }

        $this->data['totals'] = $total_data;

        if (isset($this->session->data['shipping_method']['title'])) {
            $this->data['shipping_method'] = $this->session->data['shipping_
method']['title'];
        } else {
                $this->data['shipping_method'] = $this->customer->
getShippingMethodTitle();
        }

        if (isset($this->session->data['payment_method']['title'])) {
            $this->data['payment_method'] = $this->session->data['payment_method
']['title'];
        } else {
            $this->data['payment_method'] = $this->customer->getPaymentMethodTitle();
        }

        if (file_exists(DIR_TEMPLATE . $this->config->get('config_template') . '/template/
```

```
checkout/confirm.tpl')) {
                    $this->template = $this->config->get('config_template') . '/template/
checkout/confirm.tpl';
                } else {
                    $this->template = 'default/template/checkout/confirm.tpl';
                }

            }

            return $this->render();
        }


        public function update() {
            if ((!$this->cart->hasProducts() && (!isset($this->session->data['vouchers']) ||
!$this->session->data['vouchers'])) || (!$this->cart->hasStock() && !$this->config->
get('config_stock_checkout'))) {
                $json['redirect'] = $this->url->link('checkout/cart');
            }

            $this->load->model('account/address');

            if ($this->cart->hasShipping()) {
                $this->load->model('account/address');

                if ($this->customer->isLogged()) {
                    if(isset($this->session->data['shipping_address_id']))
                    $shipping_address = $this->model_account_address->getAddress($this->
session->data['shipping_address_id']);
                    else
                    $shipping_address = $this->model_account_address->getAddress($this->
customer->getAddressId());
                } else{
                    $this->redirect($this->url->link('account/login', '', 'SSL'));
                }

                if (!isset($shipping_address)) {
                    $json['redirect'] = $this->url->link('checkout/checkout', '', 'SSL');
                }

                if (!isset($this->session->data['shipping_method'])) {
                    $json['redirect'] = $this->url->link('checkout/checkout', '', 'SSL');
                }
            } else {
                unset($this->session->data['guest']['shipping']);
                unset($this->session->data['shipping_address_id']);
                unset($this->session->data['shipping_method']);
                unset($this->session->data['shipping_methods']);
```

```
        }

        $json = array();

        if (!$json) {
            $total_data = array();
            $total = 0;
            $taxes = $this->cart->getTaxes();

            $this->load->model('setting/extension');

            $sort_order = array();

            $results = $this->model_setting_extension->getExtensions('total');

            foreach ($results as $key => $value) {
                $sort_order[$key] = $this->config->get($value['code'] . '_sort_order
');
            }

            array_multisort($sort_order, SORT_ASC, $results);

            foreach ($results as $result) {
                if ($this->config->get($result['code'] . '_status')) {
                    $this->load->model('total/' . $result['code']);

                    $this->{'model_total_' . $result['code']}->getTotal($total_data,
$total, $taxes);
                }
            }

            $sort_order = array();

            foreach ($total_data as $key => $value) {
                $sort_order[$key] = $value['sort_order'];
                $this->log->debug('IlexDebug:: get total data '. $value['code']);
            }

            array_multisort($sort_order, SORT_ASC, $total_data);

            $this->language->load('checkout/checkout');

            $data = array();

            $data['invoice_prefix'] = $this->config->get('config_invoice_prefix');
            $data['store_id'] = $this->config->get('config_store_id');
            $data['store_name'] = $this->config->get('config_name');
```

```
if ( $ data['store_id']) {
    $ data['store_url'] = $ this -> config -> get('config_url');
} else {
    $ data['store_url'] = HTTP_SERVER;
}

if ( $ this -> customer -> isLogged()) {
    $ data['customer_id'] = $ this -> customer -> getId();
    $ data['customer_group_id'] = $ this -> customer -> getCustomerGroupId();
    $ data['firstname'] = $ this -> customer -> getFirstName();
    $ data['lastname'] = $ this -> customer -> getLastName();
    $ data['email'] = $ this -> customer -> getEmail();
    $ data['telephone'] = $ this -> customer -> getTelephone();
    $ data['fax'] = $ this -> customer -> getFax();

    $ this -> load -> model('account/address');
    if(isset( $ this -> session -> data['payment_address_id'])){
        $ payment_address = $ this -> model_account_address -> getAddress
( $ this -> session -> data['payment_address_id']);
    }
} elseif (isset( $ this -> session -> data['guest'])) {
    $ data['customer_id'] = 0;
    $ data['customer_group_id'] = $ this -> config -> get('config_customer_
group_id');
    $ data['firstname'] = $ this -> session -> data['guest']['firstname'];
    $ data['lastname'] = $ this -> session -> data['guest']['lastname'];
    $ data['email'] = $ this -> session -> data['guest']['email'];
    $ data['telephone'] = $ this -> session -> data['guest']['telephone'];
    $ data['fax'] = $ this -> session -> data['guest']['fax'];

    $ payment_address = $ this -> session -> data['guest']['payment'];
}

if(isset( $ payment_address)){
    $ data['payment_firstname'] = $ payment_address['firstname'];
    $ data['payment_lastname'] = $ payment_address['lastname'];
    $ data['payment_company'] = $ payment_address['company'];
    $ data['payment_address_1'] = $ payment_address['address_1'];
    $ data['payment_address_2'] = $ payment_address['address_2'];
    $ data['payment_city'] = $ payment_address['city'];
    $ data['payment_postcode'] = $ payment_address['postcode'];
    $ data['payment_zone'] = $ payment_address['zone'];
    $ data['payment_zone_id'] = $ payment_address['zone_id'];
    $ data['payment_country'] = $ payment_address['country'];
    $ data['payment_country_id'] = $ payment_address['country_id'];
    $ data['payment_address_format'] = $ payment_address['address_format'];
}else{
    $ data['payment_firstname'] = '';
```

```
            $ data['payment_lastname'] = '';
            $ data['payment_company'] = '';
            $ data['payment_address_1'] = '';
            $ data['payment_address_2'] = '';
            $ data['payment_city'] = '';
            $ data['payment_postcode'] = '';
            $ data['payment_zone'] = '';
            $ data['payment_zone_id'] = '';
            $ data['payment_country'] = '';
            $ data['payment_country_id'] = '';
            $ data['payment_address_format'] = '';
        }

        if (isset( $ this -> session -> data['payment_method']['title'])) {
             $ data['payment_method'] = $ this -> session -> data['payment_method']
['title'];
        } else {
            $ data['payment_method'] = $ this -> customer -> getPaymentMethodTitle();
        }

        if ( $ this -> cart -> hasShipping()) {
            if ( $ this -> customer -> isLogged()) {
                $ this -> load -> model('account/address');
                if(isset( $ this -> session -> data['shipping_address_id']))
                 $ shipping_address = $ this -> model_account_address -> getAddress
( $ this -> session -> data['shipping_address_id']);
                else
                 $ shipping_address = $ this -> model_account_address -> getAddress
( $ this -> customer -> getAddressId());

            } else{
                $ this -> redirect( $ this -> url -> link('account/login', '', 'SSL'));
            }

            $ data['shipping_firstname'] = $ shipping_address['firstname'];
            $ data['shipping_lastname'] = $ shipping_address['lastname'];
            $ data['shipping_mobile'] = $ shipping_address['mobile'];
            $ data['shipping_phone'] = $ shipping_address['phone'];
            $ data['shipping_company'] = $ shipping_address['company'];
            $ data['shipping_address_1'] = $ shipping_address['address_1'];
            $ data['shipping_address_2'] = $ shipping_address['address_2'];
            $ data['shipping_city'] = $ shipping_address['city'];
            $ data['shipping_postcode'] = $ shipping_address['postcode'];
            $ data['shipping_zone'] = $ shipping_address['zone'];
            $ data['shipping_zone_id'] = $ shipping_address['zone_id'];
            $ data['shipping_country'] = $ shipping_address['country'];
            $ data['shipping_country_id'] = $ shipping_address['country_id'];
            $ data['shipping_address_format'] = $ shipping_address['address_format'];
```

```
            if (isset( $ this -> session -> data[ 'shipping_method'][ 'title'])) {
                $ data[ 'shipping_method'] = $ this -> session -> data[ 'shipping_method']
[ 'title'];
            } else {
                $ data[ 'shipping_method'] = $ this -> customer -> getShippingMethodTitle();
            }
        } else {
            $ data[ 'shipping_firstname'] = '';
            $ data[ 'shipping_lastname'] = '';
            $ data[ 'shipping_mobile'] = '';
            $ data[ 'shipping_phone'] = '';
            $ data[ 'shipping_company'] = '';
            $ data[ 'shipping_address_1'] = '';
            $ data[ 'shipping_address_2'] = '';
            $ data[ 'shipping_city'] = '';
            $ data[ 'shipping_postcode'] = '';
            $ data[ 'shipping_zone'] = '';
            $ data[ 'shipping_zone_id'] = '';
            $ data[ 'shipping_country'] = '';
            $ data[ 'shipping_country_id'] = '';
            $ data[ 'shipping_address_format'] = '';
            $ data[ 'shipping_method'] = '';
        }

        if ( $ this -> cart -> hasShipping()) {
             $ this -> tax -> setZone( $ shipping_address[ 'country_id'], $ shipping_
address[ 'zone_id']);
        } else {
            $ this -> tax -> setZone( $ payment_address[ 'country_id'], $ payment_address
[ 'zone_id']);
        }

        $ product_data = array();

        foreach ( $ this -> cart -> getProducts() as $ product) {
            $ option_data = array();

            foreach ( $ product[ 'option'] as $ option) {
                if ( $ option[ 'type'] != 'file') {
                    $ option_data[ ] = array(
                                   'product_option_id'       => $ option[ 'product_
option_id'],
                                   'product_option_value_id' => $ option[ 'product_
option_value_id'],
                                   'product_option_id'       => $ option[ 'product_
option_id'],
                                   'product_option_value_id' => $ option[ 'product_
```

```
option_value_id'],
                                        'option_id'            => $option['option_id'],
                                        'option_value_id'      => $option['option_
value_id'],
                                        'name'                 => $option['name'],
                                        'value'                => $option['option_
value'],
                                        'type'                 => $option['type']
                    );
                } else {
                    $this->load->library('encryption');

                    $encryption = new Encryption($this->config->get('config_
encryption'));

                    $option_data[] = array(
                                        'product_option_id'    => $option['product_
option_id'],
                                        'product_option_value_id' => $option['product_
option_value_id'],
                                        'product_option_id'    => $option['product_
option_id'],
                                        'product_option_value_id' => $option['product_
option_value_id'],
                                        'option_id'            => $option['option_id'],
                                        'option_value_id'      => $option['option_
value_id'],
                                        'name'                 => $option['name'],
                                        'value'                => $encryption->
decrypt($option['option_value']),
                                        'type'                 => $option['type']
                    );
                }
            }

            $product_data[] = array(
                                'product_id' => $product['product_id'],
                                'href' => $this->url->link('product/product', '&product_
id='. $product['product_id']),
                                'name'       => $product['name'],
                                'model'      => $product['model'],
                                'option'     => $option_data,
                                'download'   => $product['download'],
                                'quantity'   => $product['quantity'],
                                'subtract'   => $product['subtract'],
                                'price'      => $product['price'],
                                'total'      => $product['total'],
                                'tax'        => $this->tax->getRate($product['tax_
```

```
class_id'])
                );
            }

            // Gift Voucher
            if (isset($this->session->data['vouchers']) && $this->session->data
['vouchers']) {
                foreach ($this->session->data['vouchers'] as $voucher) {
                    $product_data[] = array(
                                'product_id' => 0,
                                'href'       => '#',
                                'name'       => $voucher['description'],
                                'model'      => '',
                                'option'     => array(),
                                'download'   => array(),
                                'quantity'   => 1,
                                'subtract'   => false,
                                'price'      => $voucher['amount'],
                                'total'      => $voucher['amount'],
                                'tax'        => 0
                    );
                }
            }

            $data['products'] = $product_data;
            $this->data['products'] = $product_data;

            $data['totals'] = $total_data;

            if(isset($this->session->data['comment'])){
                $data['comment'] = $this->session->data['comment'];
            }else{
                $data['comment'] = '';
            }
            $data['total'] = $total;
            $data['reward'] = $this->cart->getTotalRewardPoints();

            if (isset($this->request->cookie['tracking'])) {
                $this->load->model('affiliate/affiliate');

                    $affiliate_info = $this->model_affiliate_affiliate->
getAffiliateByCode($this->request->cookie['tracking']);

                if ($affiliate_info) {
                    $data['affiliate_id'] = $affiliate_info['affiliate_id'];
                    $data['commission'] = ($total / 100) * $affiliate_info['commission'];
                } else {
                    $data['affiliate_id'] = 0;
                    $data['commission'] = 0;
                }
            } else {
```

```
            $ data['affiliate_id'] = 0;
            $ data['commission'] = 0;
        }

        $ data['language_id'] = $ this -> config -> get('config_language_id');
        $ data['currency_id'] = $ this -> currency -> getId();
        $ data['currency_code'] = $ this -> currency -> getCode();
        $ data['currency_value'] = $ this -> currency -> getValue( $ this -> currency ->
getCode());
        $ data['ip'] = $ this -> request -> server['REMOTE_ADDR'];

        $ this -> load -> model('checkout/order');

        if (isset( $ this -> session -> data['payment_method'])) {
            $ this -> data['payment_code'] = $ this -> session -> data['payment_method']
['code'];
            $ this -> customer -> setPaymentMethod( $ this -> data['payment_code']);
            $ this -> log -> debug('IlexDebug:: Checkout comfirm payment_method '. $ this ->
data['payment_code']);
        } else if( $ this -> customer -> getPaymentMethod()!= ''){
            $ this -> data['payment_code'] = $ this -> customer -> getPaymentMethod();
            $ this -> log -> debug('IlexDebug:: Checkout comfirm 2 getPaymentMethod '.
 $ this -> data['payment_code']);
        } else{

            $ this -> data['payment_code'] = $ this -> config -> get('config_default_
payment');
            $ this -> log -> debug('IlexDebug:: Checkout comfirm 3 '. $ this -> data
['payment_code']);
        }

        $ data['payment_code'] = $ this -> data['payment_code'];

        if(! isset( $ this -> session -> data['order_id']))
        $ this -> session -> data['order_id'] = $ this -> model_checkout_order -> create
( $ data);
        else
        $ this -> model_checkout_order -> modify( $ data);

        // Gift Voucher
        if (isset( $ this -> session -> data['vouchers']) && is_array( $ this -> session ->
data['vouchers'])) {
            $ this -> load -> model('checkout/voucher');

            foreach ( $ this -> session -> data['vouchers'] as $ voucher) {
                $ this -> model_checkout_voucher -> addVoucher( $ this -> session ->
data['order_id'], $ voucher);
            }
        }

        // Gift Voucher
```

```
            $this->data['vouchers'] = array();

            if (isset($this->session->data['vouchers']) && $this->session->data
['vouchers']) {
                foreach ($this->session->data['vouchers'] as $key => $voucher) {
                    $this->data['vouchers'][] = array(
                        'description' => $voucher['description'],
                        'amount'      => $this->currency->format($voucher
['amount'])
                    );
                }
            }

            $this->data['totals'] = $total_data;

            if (isset($this->session->data['shipping_method']['title'])) {
                $this->data['shipping_method'] = $this->session->data['shipping_
method']['title'];
            } else {
                $this->data['shipping_method'] = $this->customer->
getShippingMethodTitle();
            }

            if (isset($this->session->data['payment_method']['title'])) {
                $this->data['payment_method'] = $this->session->data['payment_method
']['title'];
            } else {
                $this->data['payment_method'] = $this->customer->
getPaymentMethodTitle();
            }


        }

        $this->load->library('json');

        $this->response->setOutput(Json::encode($json));
    }


    public function validate() {
        $this->load_language('checkout/checkout');

        if (!$this->customer->isLogged()) {
            $this->session->data['redirect'] = $this->url->link('checkout/checkout',
'', 'SSL');

            $this->redirect($this->url->link('account/login', '', 'SSL'));
        }

        if($this->request->get['token'] == ''){
            return true;
        }
```

```
$ json = array();

if(! isset( $ this -> session -> data[ 'payment_method']))
$ json[ 'error'][ 'warning'] = $ this -> language -> get( 'error_payment');

if( $ this -> cart -> hasShipping()){
    if(! isset( $ this -> session -> data[ 'shipping_method']))
    $ json[ 'error'][ 'warning'] = $ this -> language -> get( 'error_shipping');


    if(! isset( $ this -> session -> data[ 'shipping_address_id']))
    $ json[ 'error'][ 'warning'] = $ this -> language -> get( 'error_address');
}

if(! isset( $ json[ 'error'])){
    $ this -> load -> model( 'checkout/order');
    // added comment to order
    / * if ( $ this -> request -> server[ 'REQUEST_METHOD'] == 'POST') {
    $ this -> model_checkout_order -> updateOrderComment( $ this -> session -> data
[ 'order_id'], $ this -> request -> post[ 'comment']);
    }
    * /
    //TODO update order status to nonpay

    if( $ this -> request -> get[ 'token'] == $ this -> session -> data[ 'checkout_token']){
        if(isset( $ this -> session -> data[ 'order_id'])){

            if(isset( $ this -> session -> data[ 'payment_method'][ 'code'])){

                if(! in_array( $ this -> session -> data[ 'payment_method'][ 'code'],
$ this -> direct_payments)){
                    $ this -> log -> debug ( ' IlexDebug:: Checkout Update
updateOrderStatus() : order '. $ this -> session -> data[ 'order_id'] . ' payment_method '. $ this ->
session -> data[ 'payment_method'][ 'code']);

                $ this -> model_checkout_order -> updateOrderStatus( $ this ->
session -> data[ 'order_id'], $ this -> config -> get( 'config_order_nopay_status_id'));
                $ this -> cart -> clear();
                unset( $ this -> session -> data[ 'shipping_method']);
                unset( $ this -> session -> data[ 'shipping_methods']);
                unset( $ this -> session -> data[ 'payment_method']);
                unset( $ this -> session -> data[ 'payment_methods']);
                unset( $ this -> session -> data[ 'guest']);
                unset( $ this -> session -> data[ 'comment']);
                unset( $ this -> session -> data[ 'order_id']);
                unset( $ this -> session -> data[ 'coupon']);
                unset( $ this -> session -> data[ 'voucher']);
                unset( $ this -> session -> data[ 'vouchers']);
                unset( $ this -> session -> data[ 'order_id']);
                unset( $ this -> session -> data[ 'checkout_token']);
                }
            }
        }
```

```
            }
        }
        $this->load->library('json');
        $this->response->setOutput(Json::encode($json));
    }
```

下单购买页面的效果如图 12.7 所示。

图 12.7 下单购买页面效果

小结

本章以网上商城的设计分析与代码实现为例，重点展示了PHP动态网站的制作过程。由于篇幅所限，本章仅概要地演示了关于网上商城的几个关键业务逻辑实现。在实际开发类似网上商城的大型应用系统时，遇到的情况要复杂得多，除了实现基本的业务逻辑之外，还需要综合考虑安全、性能、容错等诸多因素。但相信通过本章的学习，读者可以触类旁通，把自己所学的知识真正应用到工作实践中去。

习题

1. 综合运用所学的知识，开发一个大学在线选课系统。
2. 综合运用所学的知识，开发一个图书借阅系统。

参 考 文 献

1. 明日科技. HTML5 从入门到精通. 北京:清华大学出版社,2012.
2. 吴丰,丁欣. Dreamweaver CS5 网页设计与制作. 北京:清华大学出版社,2012.
3. 张明真,汪可. Adobe Dreamweaver CS4 标准培训教材. 北京:人民邮电出版社,2009.
4. 曾顺. 精通 CSS+DIV 网页样式与布局. 北京:人民邮电出版社,2007.
5. 刘增杰,刘海松. 精通 DIV+CSS3 网页布局与样式. 北京:清华大学出版社,2012.
6. 王珊萨,师煊. 数据库系统概论. 4 版. 北京:高等教育出版社,2009.
7. 郑阿奇. MySQL 实用教程. 北京:电子工业出版社,2012.
8. 李慧. PHP 入门经典. 北京:机械工业出版社,2013.
9. 曾俊国,等. PHP Web 开发实用教程. 北京:清华大学出版社,2011.
10. 孔祥盛. PHP 编程基础与实例教程. 北京:人民邮电出版社,2011.